AFIT/GE/ENG/93D

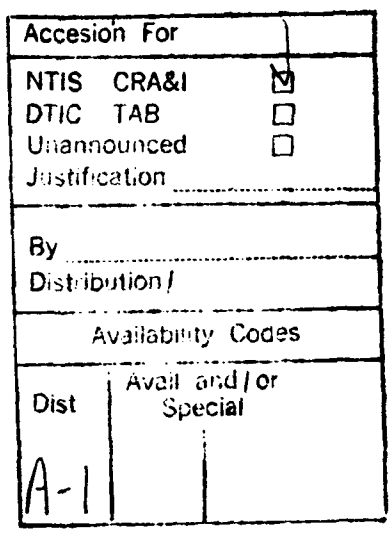
ANALYSIS AND SIMULATION OF A GPS RECEIVER DESIGN USING COMBINED DELAY-LOCK AND MODIFIED TANLOCK LOOPS

THESIS

George D. Harris, Captain, USAF

AFIT/GE/ENG/93D

The views expressed in this thesis are those of the author and do not reflect the official policy of position of the Department of Defense or the U. S. Government.



AFIT/GE/ENG/93D-13

ANALYSIS AND SIMULATION OF A GPS RECEIVER DESIGN USING COMBINED DELAY-LOCK AND MODIFIED TANLOCK LOOPS

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

George D. Harris, B.S.E.E.

Captain, USAF

December 1993

Acknowledgements

I would like to extend my sincerest thanks to the many people who helped me complete this research. First of all, I wish to thank my study partners, Mike, Dennis, Mark, Tony, and Lynn, who helped me, immensely, to understand the material in the many classes we shared. I want to thank my advisors, Major Mark Mehalic, Captain Joe Sacchini, and Lt Col Robert Riggins, for their patience and insight. Finally, I want to thank my loving wife, Janie, for the constant support and encouragement she provided during the 19 month separation we endured while I completed my studies.

Table of Contents

List of Figures

Figure — Page

Symbol		Introduced on Page
$D_g(\tau_e/\Delta)$	Autocorrelation difference function, $R_{g+}(\tau_e/\Delta) - R_{g-}(\tau_e/\Delta)$	14
$F(p)$	Filter transfer function using the Heaviside operator	16
G_1, G_2	Loop filter control parameters	16
$g(t)$	Spreading pseudo-noise (PN) sequence	9
K_c	Voltage controlled clock gain	16
K_g, K_m	Loop mixer gains	12
K_0	Voltage controlled oscillator gain	43
M	Number of chips in one period of g(t)	11
$m(t)$	Message signal	11
$N_c(t)$, $N_s(t)$	Additive white Gaussian noise components of input noise	30
N_0	Input noise power spectral density (One-sided)	30
$n(t)$	Additive white Gaussian noise on the received signal	29
$n_e(t)$	Phase error noise	51
P	Input signal power	11
P_v	Power of VCO output signal	41
$R_{g+}(\tau_e/\Delta)$	Autocorrelation function of PN sequence offset by $+\Delta/2$	12
$R_{g-}(\tau_e/\Delta)$	Autocorrelation function of PN sequence offset by $-\Delta/2$	13
$S_g(\tau_e/\Delta)$	Autocorrelation summation function, $R_{g+}(\tau_e/\Delta) + R_{g-}(\tau_e/\Delta)$	16
$S_n(\omega)$	Noise power spectral density	32
s	Laplace Transform variable $s = p = j\omega$	16
$s(t)$	Loop input signal	11
T	Chip duration in seconds	11
W_L	Two-sided loop bandwidth in rads/sec	17
$\Delta\omega_v$	Offset between received signal and VCO output frequency	42
$\Delta/2$	Delay-lock loop offset (normally equal to T/2)	10
$\Theta(t)$	Phase of received signal	11
$\Theta_e(t)$	Loop phase error, $\Theta(t) - \Theta_v(t)$	42
$\Theta_{en}(t)$	Phase error due to noise	52
$\Theta_v(t)$	Phase of VCO output signal	41
θ_0	Arbitrary phase of transmitted signal	11
σ_{en}^2	Noise error variance	33
τ	Transmission delay of satellite signal	9
$\hat{\tau}$	Transmission delay estimate	9
$\hat{\tau}/\Delta$	Normalized transmission delay estimate	16
τ_e	Transmission delay estimate error, $(\tau - \hat{\tau})$	12
$\hat{\tau}_e/\Delta$	Normalized transmission delay estimate error	12
τ_{en}	Delay error due to noise	33
ω	Actual frequency of carrier at reception	11
ω_0	Nominal carrier frequency	11

Abbreviation		Introduced on Page
AWGN	Additive White Gaussian Noise	7
BPF	Bandpass Filter	10
BPSK	Binary Phase-Shift Keying	9
C/A	Clear Acquisition	2
DLL	Delay-Lock Loop	5
DSP	Digital Signal Processing	4
DS/SS	Direct Sequence/Spread Spectrum	1
FFT	Fast Fourier Transform	5
GPS	Global Positioning System	1
HOW	Hand-Over-Word	3
MCTL	Modified Code Tracking Loop	6
MTLL	Modified Tanlock Loop	6
P-code	Precision Code	2
PLL	Phase Lock Loop	39
PN	Pseudo-Random Noise	2
PSD	Power Spectral Density	21
SNR	Signal-to-Noise Ratio	33
SPW™	Signal Processing Workstation®	8
TLL	Tanlock Loop	6
TTFF	Time To First Fix	2
VCC	Voltage Controlled Clock	9
VCO	Voltage Controlled Oscillator	39

Abstract

The purpose of this thesis was to investigate the performance of two types of tracking loops used in Global Positioning System (GPS) receivers. The first loop, the Delay-Lock Loop (DLL), is responsible for maintaining synchronization with the received PN sequence. Synchronization of the PN sequence is essential for despreading the direct-sequence spread spectrum (DS/SS) broadcast and demodulating the transmitted data. The second loop, the Modified Tanlock Loop (MTLL), is responsible for maintaining synchronization with the carrier signal. Carrier synchronization is essential for optimum data demodulation. The performance of the two loops is investigated first separately then their performance is evaluated when operated together. This thesis is an investigation on the ability of these two loops to overcome corruption of the input signal due to noise. Expanding the dynamic operating range of these loops can significantly improve GPS receiver operation. Results indicate the performance of the loops was better than theoretical predictions by maintaining lock across a wide range of loop gains and SNRs. However, when the loops were combined, the loops did not perform as predicted by theory. All simulations display phenomena which was not present in the theoretical predictions.

ANALYSIS AND SIMULATION OF A GPS RECEIVER DESIGN USING COMBINED DELAY-LOCK AND MODIFIED TANLOCK LOOPS

I. INTRODUCTION

1.1 BACKGROUND

The NAVSTAR Global Positioning System (GPS) is a satellite-based radio navigation system. GPS receivers determine their position and velocity by measuring the time delay between signals of at least three satellites--a fourth satellite will eliminate any time inaccuracies between satellite and receiver clocks. The receiver calculates a navigation solution through triangulation of the signals. The GPS system is a Direct-Sequence Spread Spectrum (DS/SS) based system which spreads the signal energy across a wide spectrum. Upon locating the signal, the receiver despreads the spectrum, locks onto the signal, and recovers the navigation data.

When a GPS receiver detects a DS/SS signal, the signal power is spread out over such a large bandwidth that the signal is below the thermal noise level. When the satellite signal is multiplied with a properly synchronized receiver generated code, the resulting signal collapses in frequency into the original carrier band. Thus, the signal power is again concentrated into a narrow frequency band and well above thermal noise level. A DS/SS system is also an antijam system which has large instantaneous bandwidth, provides an excellent mechanism for ranging, has a low probability of intercept, improved antenna tracking capability and SNR performance, and is less affected by multipath transmissions than other signal recovery systems.

Each satellite transmits on two L-band carrier frequencies (L1 and L2). L1 is the primary frequency at 1575.42 MHz and L2 is the secondary frequency at 1227.6 MHz. They enable the

receiver to perform frequency compensation for signal delay due to ionospheric refraction. Since each satellite transmits on the same two modulation frequencies, two unique pseudo-random noise (PN) codes are assigned to each satellite. The unique PN codes enable a receiver to distinctively discriminate among the multiple satellite signals it receives and select only those signals it needs to use for navigation, even though all the satellites operate on the same two frequencies. The two codes are a Clear/Acquisition (C/A) and a Precision (P) code.

The P-code is a 267-day long non-repetitive code sequence and each satellite is assigned a unique one week segment transmitted on both L1 and L2. The C/A code is a 1023-bit long PN code with a clock rate of 1.023 MHz; therefore, it takes only 1 ms to run through the entire code. The C/A code is used to assist the receiver in reducing the time to acquire the longer P-code. Both code signals are biphase modulated using +1's and -1's only (binary data) at 50 bps. The L1 carrier is phase modulated by both PN signals with the C/A code lagging the P-code by 90 degrees and L2 is modulated with C/A code only. The transmitted signal is delayed due to the path length between satellite and receiver, Doppler-shifted due to relative velocity, and attenuated. Since the receiver has a different clock than the satellite, if it observes the signal from only one satellite, it cannot tell the difference between its clock offset and the signal delay due to range. A receiver is defined by its satellite signal tracking type. There are basically two different types of GPS receivers: continuous tracking and sequential tracking.

For continuous tracking, each receiver channel is dedicated to one satellite; therefore, at least four channels are required to solve for three unknown position components and time. A fifth channel can perform dual frequency measurements for ionospheric delay, determine interchannel bias correction, and keep track of all other satellites in view to select the next satellite to track in a changing constellation. A five channel set provides the greatest accuracy under high dynamics,

the maximum antijam capability, and the lowest time to first fix (TTFF). TTFF is the elapsed time it takes (after a set has been turned on for the GPS standard of 7 minutes) to display position, velocity, and time with specified accuracy. A five channel set is ideal for high dynamic vehicles, such as fighter aircraft, and submarines (due to its low TTFF), or any vehicles requiring antijamming capabilities. In sequential tracking, the receiver channel switches through satellites and tracks one satellite at a time. The advantage is a cheaper, smaller, and lighter receiver that requires less power than a continuous tracking set. The disadvantage is poor dynamic tracking, poor jamming immunity, and longer acquisition time. The receiver determines the satellites available for tracking and seeks a satellite C/A code using satellite position estimates residing in memory and the user's approximate present position, velocity, and time. If no stored almanac information exists, or the estimates of position and time are poor, the receiver will go into a search-sky mode and attempt to lock on to any satellite in view. When the receiver tracks the first satellite, it demodulates the navigation message and reads almanac information about all other satellites in the constellation. If a receiver platform is equipped with a precise time reference (atomic clock), the receiver can acquire the P-code directly without first using the C/A code. This faster acquisition of the P-code is valuable for submarines since it reduces their surface time.

The phase offset between a receiver and incoming code is directly proportional to a pseudorange. It is called a pseudorange since the receiver is measuring the code delay time with an imprecise clock and a bias of "fixed" magnitude included in each range estimate. At the beginning of tracking, the satellite code will not correlate with the reference code. This is due to the time delay for a satellite signal to reach the receiver and a receiver clock offset. Therefore, the reference code is shifted until it achieves maximum correlation. The magnitude of the shift determines the value of the pseudorange. After synchronization and signal lock are achieved, the

satellite's navigation data is demodulated from the carrier signal to obtain data for accurate pseudorange calculations. The data also contains the satellite's ephemeris (orbital parameters) and clock information which the receiver uses to calculate a navigation solution. In addition, the navigation message contains a Hand-Over-Word (HOW) to enable the receiver to obtain P-code phase information for the transfer of C/A code to P-code.

Since the received satellite signal level near the earth is less than background noise level, the receiver's acquisition loop uses a correlation technique to recover the signal. Then, carrier and code tracking loops work together in an iterative process to acquire and track signals. The carrier tracking loop compares a locally-generated L1 or L2 frequency with the received signal. The loop then adjusts the receiver-generated frequency until it matches the incoming frequency, thereby determining the relative velocity between receiver and satellite. The code tracking loop generates and stores a replica of each satellite's C/A PN code (or P-code), with an estimated ranging delay, which it uses to synchronize with a specific satellite. In order to match the received signal with its stored replica, the two signals must have the same center frequency and phase. The center frequency of the replica is set using a Doppler estimate from the carrier loop.

Each GPS receiver design depends on its specific application. A GPS receiver is used for many applications, for example tracking tectonic plate movement and aircraft navigation. Tracking tectonic plate movement requires millimeter accuracy, but is time independent; however, fighter aircraft require rapid position updates, but needs only meter accuracy.

1.2 Problem Statement

Coherent jamming and multi-path interference cause serious concern to DOD personnel in GPS receiver operations. Current GPS receiver models do not allow for the use of recent

innovations in DS/SS and Digital Signal Processing (DSP) techniques. Each GPS receiver contains an acquisition loop, a carrier tracking loop, and a code tracking loop. First, the acquisition loop searches and achieves initial phase lock with the incoming navigation signal. Next, a carrier tracking loop acquires and maintains phase synchronization of the carrier signal. Finally, a code tracking loop maintains bit synchronization with the data on the carrier signal for demodulation.

Since all three loops must be present in any GPS receiver design, computer modeling all three prior to the construction of a digital receiver is warranted. With computer modeling of a GPS receiver, different designs containing different phase-locked loops can be simulated to design a high dynamic receiver. Models could be enhanced as future proposed loop modifications or new digital communication techniques are introduced. Models could be tested and analyzed prior to any hardware implementation.

In 1992, Schmitz successfully modeled an acquisition loop using a Fast Fourier Transform (FFT) to synchronize an incoming DS/SS signal with a locally generated reference signal [SCH92]. Hird modeled a code tracking and a carrier tracking loop using a Delay-Lock Loop and a modified Tanlock Loop respectively; however, the simulation results failed to confirm theoretical results [HIR92]. This thesis will focus on the code and carrier tracking loop models and solve the problems associated with Hird's work.

1.3 Summary of Current Knowledge

A Delay-Lock Loop (DLL) is a code tracking loop used to track the delay difference between an incoming PN sequence correlated with an early and late version of its stored replica. In 1966, Gill analyzed the performance of the DLL for use with radio frequencies [GIL66]. He

investigated amplitude and phase modulation together with phase coherent correlation, envelope correlation, and phase lock modulation with video correlation. Subsequent papers on the subject proposed modifications to the DLL; however, each modification could not overcome sensitivity to gain imbalance in the loop. Other modifications introduced excessive jitter and noise.

In 1980, Yost and Boyd proposed a modified code tracking loop (MCTL) to counter the gain imbalance in the loop; however, noise and jitter were still a problem [YOS80][YOS82]. Even though the MCTL could not solve all the problems, it displays a superior tracking ability compared to the earlier DLLs. New digital modifications were proposed which eliminates the effects of gain imbalance and steady-state jitter [GAU91].

A Tanlock Loop (TLL) is a carrier tracking loop used to track a satellite's carrier signal. Due to the loop's large linear operating range, a TLL can track a signal with high dynamics and a low signal-to-noise ratio. The difference between the phase of the incoming signal and a voltage controlled oscillator in the receiver is called the phase error. The TLL can maintain linear operation with phase errors up to 157 degrees [ROB62]. In 1964, Balodis demonstrated that a TLL in place of a phase-lock loop improved threshold reception by 2 to 6 dB [BAL64].

In 1982, Lee and Un proposed a modified TLL (MTLL) which replaced the tangent phase detector in the loop with an arctangent device [LEE82]. This modification increased the linear operating range of the loop to approximately 180 degrees. The MTLL also provided insensitivity to input signal power which effectively eliminates the need for automatic gain control circuitry. With its increased noise immunity, the loop also provided the ability to analyze loop performance with linear difference equations instead of linear approximations. In 1987, Cho and Un demonstrated that by increasing the sampling rate the locking range of the MTLL can be extended even further than 180 degrees [CHO87].

1.4 ASSUMPTIONS

This thesis assumes that one satellite is in view for the one channel being modeled and initially no data is being transmitted. The acquisition phase is already completed and the GPS receiver is locked onto the PN code to within one-half bit (chip). Also, the signal is already converted to an intermediate frequency (IF) by a down converter and the incoming signal consists only of a C/A PN code modulated onto a carrier signal. All frequencies used in the simulation are much lower than in a nominal GPS system. It does not take into account satellite clock error, atmospheric delays, group delays due to satellite equipment, and receiver noise and resolution.

1.5 SCOPE

The two loops modeled are the current digital tracking loops being proposed currently. They consist of a Modified Tanlock Loop (MTLL) and a Delay-Lock Loop (DLL) all degraded with additive white Gaussian noise (AWGN). For the ease of simulation, the frequencies being used for the DLL and MTLL are at 10 and 25 Hz respectively; however, the same loop performance should occur at IF. Also, only five shift registers are being used instead of the nominal ten. The application design modeled is for use during high dynamic maneuvers and noisy environments--for example fighter aircraft. Manufacturing and cost constraints are not considered.

1.6 APPROACH

This thesis will present equations governing the theoretical performance of two types of tracking loops degraded with white noise. Models of code tracking and carrier tracking loops are investigated by modifying existing loops already developed. These modifications include decreasing the limits of integration on the individual loop arm filters to integrate over one chip

and one cycle respectively, routing the clock signal into the loop filters to control integration, and changing all integrators to a more stable integrator. Next, the two loops are analyzed as first and second-order loops with the signal only corrupted by AWGN. Finally, the two loops are joined together to determine the data recovery point and loop response as a working code/carrier tracking receiver design.

1.7 Materials and Equipment

Models are developed using the Signal Processing WorkSystem®(SPW™) simulation software version 3.0 by Comdisco® Systems, Inc. of Foster City, California. The software is located on a Sun Workstation® in the Communications/Radar Laboratory, room 225, building 640.

1.8 Thesis Organization

Chapter II presents an analysis of a current DLL and a description of its computer model. Chapter III summarizes the simulation results of a first and second-order DLL with the signal only degraded by AWGN. Chapter IV introduces an analysis of a current MTLL and a description of its computer model. Chapter V summarizes the simulation results of a first and second-order MTLL with the signal only degraded by AWGN. Chapter VI explores the joining of the two loops, determines the data recovery point, and summarizes the simulation results of the DLL/MTLL configuration degraded by AWGN. Finally, Chapter VII summarizes the results of the thesis and discusses limitations and recommendations for future research.

II. DELAY-LOCK LOOP SIMULATION

2.1 CHAPTER OVERVIEW

This chapter presents an overview and analysis of a Delay-Lock Loop (DLL) model. Section 2.2 presents a basic overview of a DLL. In Section 2.3, the equations developed to analyze a DLL are described. Section 2.4 introduces the DLL model and its individual components used to simulate a DLL. Finally, Section 2.5 summarizes the chapter.

2.2 DELAY-LOCK LOOP OVERVIEW

By making use of the autocorrelation property of the code, the acquisition loop of a GPS receiver performs a code alignment (to within half a chip) with a specific PN code contained in a GPS satellite's signal. Once acquisition or synchronization of the PN sequence is attained, tracking takes place. Since the exact phase and frequency of the signal is not known prior to synchronization, a noncoherent code tracking loop (in this case a DLL) is used to track the received PN sequence. A basic noncoherent DLL block diagram for a DS/SS system using binary phase shift keying (BPSK) is shown in Figure 1.

The incoming satellite signal, s(t), is fed into multipliers where it is mixed with a local replica of the PN code, which removes the PN code from the incoming signal, and despreads the spectrum. The locally generated code, g(t), of the tracking loop is offset in phase from the incoming signal by a time $\hat{\tau}$, where the $\hat{\tau}$ indicates the loop's approximation of the propagation delay, τ. The generation of the local PN sequence is controlled by a Voltage Controlled Clock (VCC). The frequency of the clock that drives the PN sequence generator is controlled by a filtered version of the loop error signal, e(t), at the output of the loop filter.

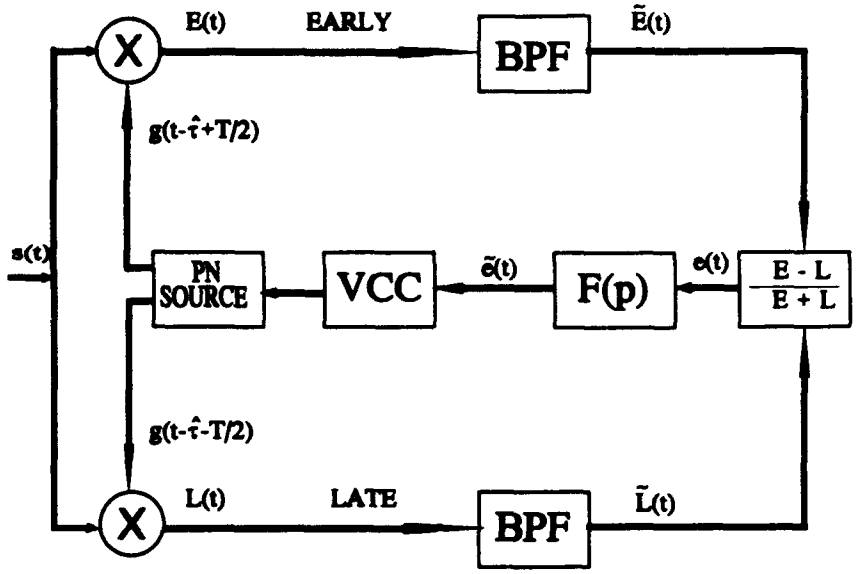


FIGURE 1 DELAY LOCK LOOP

The loop provides fine synchronization by first generating two PN sequences $g(t-\hat{\tau}+T/2)$ and $g(t-\hat{\tau}-T/2)$ delayed from each other by one chip (where $\Delta/2 = T/2$ sec and T is the chip duration). This half chip offset essentially forms "Early" and "Late" arms of the loop. The two bandpass filters (BPF) function as envelope detectors and are designed to pass the data and average the product of s(t) and the two PN sequences $g(t-\hat{\tau}+/-T/2)$; thereby reducing any unwanted loop signal processing gain. The Early and Late arms of the loop are then fed into a delay detector.

The delay detector computes the difference of the two arms and then divides the result by their sum to essentially normalize the error. Normalizing in this manner will provide an improvement in reducing noise effects [OUL84]. The resulting feedback signal, e(t), is now a corresponding error signal. The error signal is passed through a loop filter prior to entering the VCC. When e(t) is positive, the feedback signal instructs the VCC to increase its frequency, thereby forcing τ to decrease, and when e(t) is negative, it instructs the VCC to decrease, thereby

forcing τ to increase. Therefore, a positive voltage will increase the clock rate of the VCC while a negative voltage will decrease the clock rate.

2.3 DELAY-LOCK LOOP ANALYSIS

To begin the analysis of the DLL, the data, m(t), and the PN code, g(t), each modulate the carrier wave using BPSK, and in the absence of noise and interference, the received waveform, s(t), can be expressed as [HIR92]:

$$s(t) = \sqrt{P}\, g(t-\tau)m(t-\tau)\cos[\omega_0 t + \Theta(t)] \qquad (1)$$

where: P is the power in the input signal, s(t), in watts

τ is the transmission or propagation delay of the input signal in seconds

$g(t-\tau)$ is a repeating pseudo-noise (PN) sequence with values ± 1, M chips in length, each chip of T sec duration. The period of $g(t-\tau)$ is MT seconds.

$m(t-\tau)$ is a message signal with values ± 1, edge synchronized with $g(t-\tau)$.

ω_0 is the nominal carrier frequency in rad/sec

$\Theta(t) \triangleq \Delta\omega t + \theta_0$ is the channel rotation resulting from dynamics such as Doppler

$\Delta\omega \triangleq [\omega - \omega_0]$ is the offset between the nominal carrier frequency, ω_0, and the actual frequency of s(t), ω.

The local PN code replica, g(t), is generated in the feedback path with an N-stage shift register identical to that used in the transmitter for generating the code initially. A PN sequence

of period M = 2N-1 is generated. For the early arm of the DLL, s(t) is mixed with a half chip advanced version of the local generated code and the loop's estimate of the propagation delay, g(t-$\hat{\tau}$+T/2). After the mixer, the early version of the signal, E(t), is now:

$$E(t) = \sqrt{P}\,K_m\,m(t-\tau)\overline{g(t-\tau)g\left(t-\hat{\tau}+\frac{\Delta}{2}\right)}\cos[\omega_0 t + \Theta(t)] \qquad (2)$$

where K_m is the mixer gain, the overbar represents statistical expectation (autocorrelation), and the self-noise term is neglected [SPI63].

The autocorrelation function of the PN sequence, $R_{g+}(\tau_e/\Delta) \triangleq g(t-\tau)g(t-\hat{\tau}+T/2)$, for the early arm of the loop, offset by +T/2, is defined as [SIM77]:

$$R_{g+}(\tau_e/\Delta) = \begin{cases} -\frac{1}{M}, & -M \le \frac{\tau_e}{\Delta} \le -\frac{3}{2} \\ 1 + \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}+\frac{1}{2}\right), & -\frac{3}{2} \le \frac{\tau_e}{\Delta} \le -\frac{1}{2} \\ 1 - \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}+\frac{1}{2}\right), & -\frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{1}{2} \\ -\frac{1}{M}, & \frac{1}{2} \le \frac{\tau_e}{\Delta} \le M \end{cases} \qquad (3)$$

where $\tau_e \triangleq \tau - \hat{\tau}$ is the delay error between the actual propagation delay of the PN sequence and the loop's estimate of it.

After E(t) is passed through the BPF, the filtered version of the early signal, is defined as:

$$\tilde{E}(t) = \sqrt{P}K_m \tilde{m}(t-\tau)R_{g-}(\tau_e/\Delta)\cos[\omega_0 t + \Theta(t)] \tag{4}$$

where ~ represents a filtered signal.

Similarly, the filtered version of the late arm's signal is defined as:

$$\tilde{L}(t) = \sqrt{P}K_m \tilde{m}(t-\tau)R_{g-}(\tau_e/\Delta)\cos[\omega_0 t + \Theta(t)] \tag{5}$$

and the autocorrelation of the half chip delayed, -T/2, PN sequence is [SIM77]:

$$R_{g-}(\tau_e/\Delta) = \begin{cases} -\frac{1}{M}, & -M \le \frac{\tau_e}{\Delta} \le -\frac{1}{2} \\ 1 + \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}-\frac{1}{2}\right), & -\frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{1}{2} \\ 1 - \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}-\frac{1}{2}\right), & \frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{3}{2} \\ -\frac{1}{M}, & \frac{3}{2} \le \frac{\tau_e}{\Delta} \le M \end{cases} \tag{6}$$

Note $R_{g-}(\tau_e) = R_{g+}(\tau_e - T)$. Figure 2a graphs the two autocorrelation functions of the loop arms.

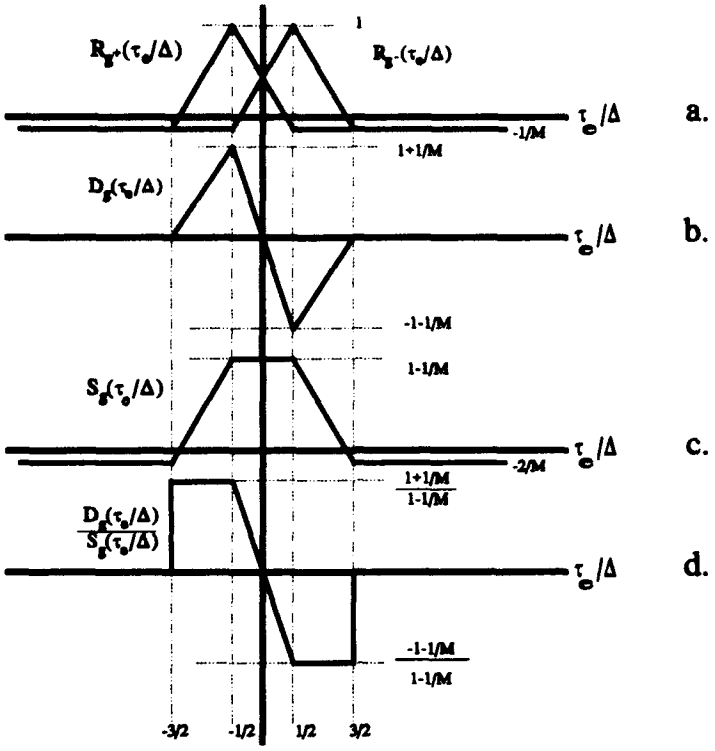


FIGURE 2 DELAY ERROR DETECTOR CHARACTERISTIC CURVES

The corresponding error signal, e(t), from the delay detector is defined as:

$$e(t) = \frac{\tilde{E} - \tilde{L}}{\tilde{E} + \tilde{L}}$$

$$= \frac{A(t) D_g(\tau_e/\Delta)}{A(t) S_g(\tau_e/\Delta)} \qquad (7)$$

$$= \frac{D_g(\tau_e/\Delta)}{S_g(\tau_e/\Delta)}$$

where: $A(t) = \sqrt{P} K_m \tilde{m}(t-\tau)\cos[\omega_0 t + \theta(t)]$

$D_g(\tau_e/\Delta) = R_{g+}(\tau_e/\Delta) - R_{g-}(\tau_e/\Delta)$

$S_g(\tau_e/\Delta) = R_{g+}(\tau_e/\Delta) + R_{g-}(\tau_e/\Delta)$

Note in the absence of noise, the ratio simplifies to the discriminator characteristic. Therefore, the error signal is only dependent on the correlation functions. Increasing the linearity of the discriminator characteristic curve increases the dynamic range of the loop.

The correlation function formed by delaying and advancing by one-half chip the PN autocorrelation functions and then taking the difference [Fig 2b] is [HIR92]:

$$D_g(\tau_e/\Delta) = \begin{cases} 0, & -M \le \frac{\tau_e}{\Delta} \le -\frac{3}{2} \\ \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}+\frac{3}{2}\right), & -\frac{3}{2} \le \frac{\tau_e}{\Delta} \le -\frac{1}{2} \\ \left(1+\frac{1}{M}\right)\frac{2\tau_e}{\Delta}, & -\frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{1}{2} \\ \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}-\frac{3}{2}\right), & \frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{3}{2} \\ 0, & \frac{3}{2} \le \frac{\tau_e}{\Delta} \le M \end{cases} \quad (8)$$

where $D_g(\tau_e/\Delta)$ is the phase detector characteristic of the PN waveform and varies linearly with t for $|\tau_e| < t$. Hence, it provides a good linear tracking region for a loop that implements $D_g(\tau_e/\Delta)$ as its equivalent error curve. The DLL achieves such an error curve by cross-correlating the received ranging signal sequence with a delayed and advanced version of the locally generated code replica.

By dividing the difference correlation function with the sum correlation function, it yields a normalized one chip linear operating region [Fig 2d].

The correlation function formed by summing the two autocorrelation functions [Fig 2c] is [HIR92]:

$$S_g(\tau_e/\Delta) = \begin{cases} -\frac{2}{M}, & -M \le \frac{\tau_e}{\Delta} \le -\frac{3}{2} \\ 2 + \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}-\frac{1}{2}\right), & -\frac{3}{2} \le \frac{\tau_e}{\Delta} \le -\frac{1}{2} \\ \left(1-\frac{1}{M}\right), & -\frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{1}{2} \\ 2 - \left(1+\frac{1}{M}\right)\left(\frac{\tau_e}{\Delta}+\frac{1}{2}\right), & \frac{1}{2} \le \frac{\tau_e}{\Delta} \le \frac{3}{2} \\ -\frac{2}{M}, & \frac{3}{2} \le \frac{\tau_e}{\Delta} \le M \end{cases} \quad (9)$$

After the error signal leaves the delay detector, it passes through a loop filter, F(p), whose transfer function defined as $F(s) = G_1 + G_2/s$. Where the Laplace Transform variable $s = j\omega = p$ and G_1 and G_2 are loop gains. Finally, the filtered error signal enters the VCC where the output is the normalized propagation delay estimation [HIR92]:

$$\frac{\hat{\tau}}{\Delta} = \int K_c \tilde{e}(t)\, dt = \frac{K_c F(p) e(t)}{p} \quad (10)$$

which drives the frequency of the PN sequence generator. K_c is the VCC gain.

The two-sided loop bandwidth is [COO86]:

$$W_L = K_c G_1 + \frac{G_2}{2G_1} \tag{11}$$

where G_1 and G_2 are loop gains in the loop filter.

2.4 DELAY-LOCK LOOP MODEL

The purpose of the DLL is to track the delay difference between two waveforms by correlating a transmitted PN sequence with a locally generated PN sequence. Figure 3 displays the DLL model used in the simulations to duplicate the tracking loop. The INPUT SIGNAL block generates a PN sequence that is output through a -6 interval delay block to simulate an "ontime" PN sequence and to account for a -1 delay within the loop. The input signal is then fed into mixers where it is multiplied with a half chip "early" and a half chip "late" version of the locally generated PN sequence. Then, the Early and Late arms of the loop are passed through INTEGRATE & DUMP blocks where a CLOCK signal controls the integration over one chip.

After the integrators, the two signals are fed into a DELAY DETECTOR where the difference of the two signals is divided by the sum of the two resulting in an error signal. Then, the error signal is passed through a LOOP FILTER, which contains another integrator. A switch block, controlled by a timer, allows the PN SOURCE block to generate and store the locally generated PN sequence prior to any added noise or interference corrupting the loop.

The PN SOURCE block generates "late", "early", and "ontime" versions of the PN sequence. The early and late versions are sent to mixers where they are mixed with the input

signal. When the DLL is locked onto the input signal, the input PN sequence is synchronized with the ontime locally generated PN sequence. Now, each block is examined in detail.

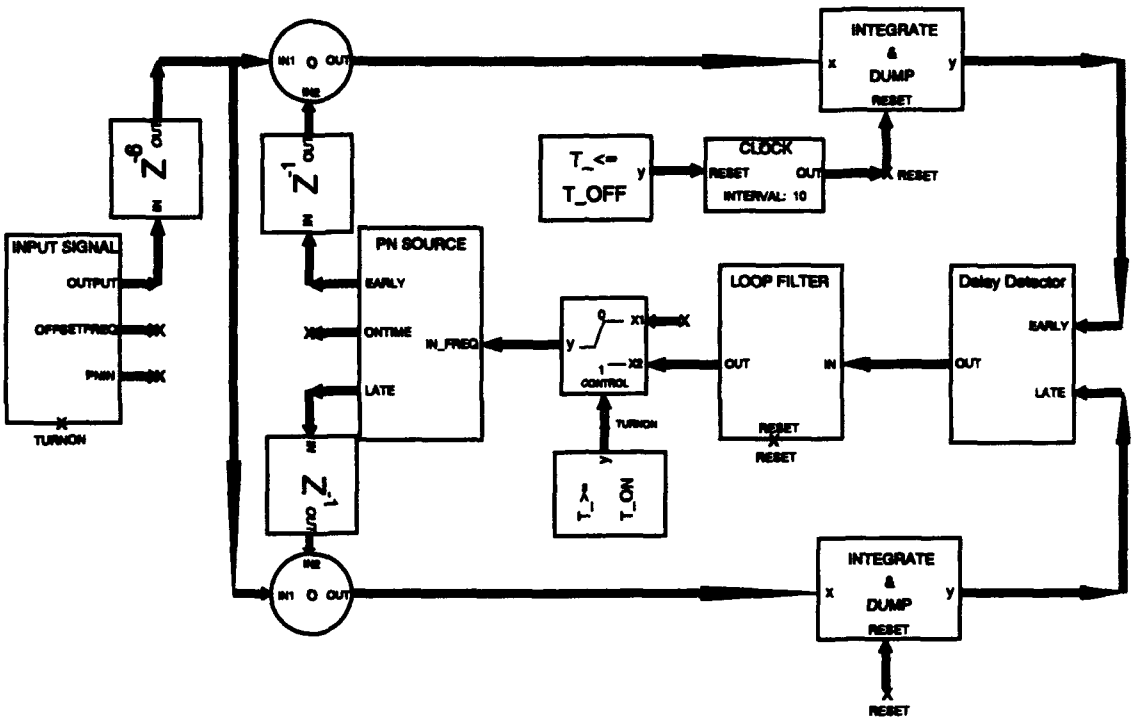


FIGURE 3 DELAY LOCK LOOP MODEL

2.4.1 INPUT SIGNAL BLOCK. Figure 4 represents the components used to generate the input signal. The purpose of this block is to generate the transmitted PN sequence. Also, noise, frequency steps, and frequency ramps can be added to the input signal from this block.

A FUNCTION GENERATOR outputs a triangle wave at a frequency of 10 Hz. By using lower frequencies, the time required to perform and analyze simulations is reduced. The triangle wave is fed through a switch and into a PROGRAMMABLE PN GENERATOR block.

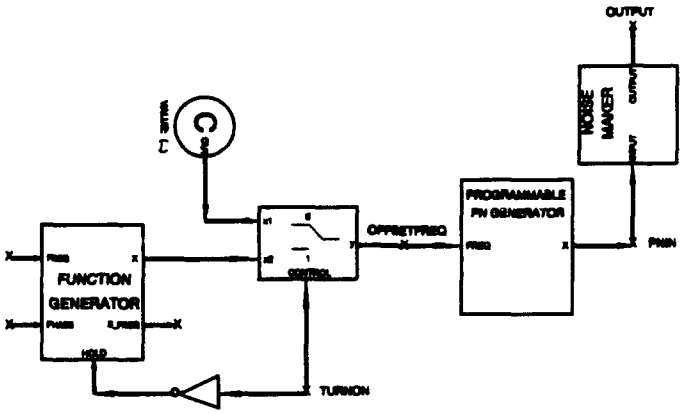


FIGURE 4 INPUT SIGNAL BLOCK

The switch is set to zero until a "turn-on" signal is issued to activate the switch and degrade the input signal by adding a frequency step or ramp. The switch allows the DLL to lock onto the incoming PN sequence prior to any Doppler shifts affecting the signal.

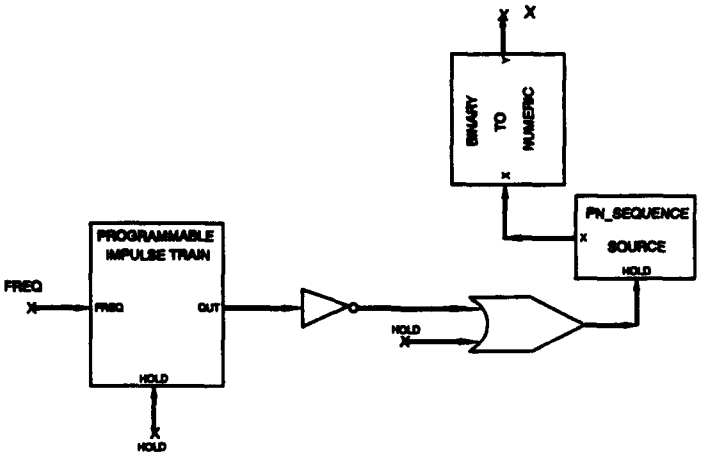


FIGURE 5 PROGRAMMABLE PN GENERATOR BLOCK

The PROGRAMMABLE PN GENERATOR block [Fig 5] contains the components necessary to generate the input PN sequence. The triangle wave first enters a PROGRAMMABLE IMPULSE TRAIN block [Fig 6] where it is fed into a FUNCTION GENERATOR. In the FUNCTION GENERATOR a square wave is generated at a frequency of 10 Hz. If the switch is never turned on, the triangle wave will not reach the FUNCTION GENERATOR and degrade the square wave.

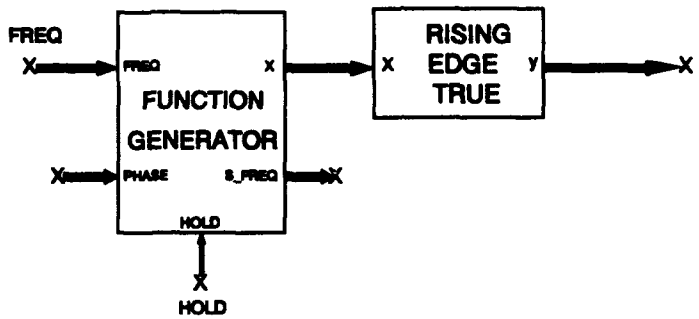


FIGURE 6 PROGRAMMABLE IMPULSE TRAIN BLOCK

Next, the square wave passes through a RISING EDGE TRUE block where it is turned into a series of impulses with a frequency of 10 Hz. The impulses output of the PROGRAMMABLE IMPULSE TRAIN block are then passed through an invertor and an OR gate where they control a PN_SEQUENCE SOURCE block. At each inverted impulse, the HOLD is pulled low and the block will output a chip.

The PN sequence contains a period of 2N-1 where N is the number of shift registers. Normally, a GPS transmitted PN sequence contains 1023 chips produced from 10 shift registers,

however, for the purposes of speeding up the simulations, the number of shift registers is set to five. This produces a PN sequence of 31 chips. Next, the PN sequence is passed through a BINARY TO NUMERIC block in order to limit the amplitude to +/- 1 volt, therefore, making the signal bipolar.

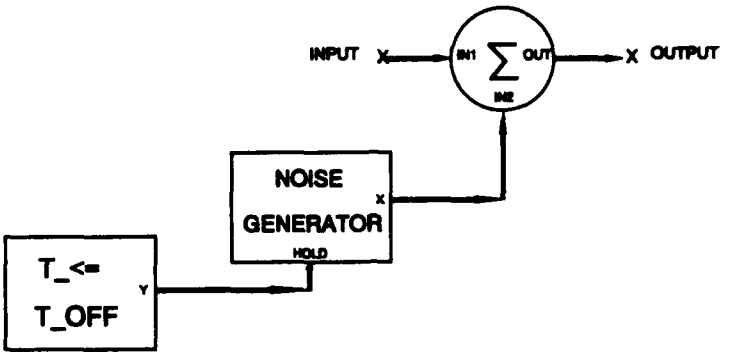


FIGURE 7 NOISE MAKER BLOCK

Finally, the PN sequence out of the PROGRAMMABLE PN GENERATOR is fed into a NOISE MAKER block [Fig 7] where a NOISE GENERATOR block generates Additive White Gaussian Noise (AWGN). The Gaussian variance of the noise, σ^2, is equal to the two-sided Power Spectral Density (PSD), $N_0/2$ watt-sec/rad. The noise is added to the input signal to generate the appropriate N_0/P sec/rad which is input to the DLL. A timer block (T_<= T_OFF) controls when the noise is added to the input signal.

2.4.2 PN SOURCE BLOCK. Figure 8 displays all the components used to generate and store the local PN sequence. A PN_SEQUENCE GENERATOR generates a PN sequence of 31 chips

and passes the sequence through a BINARY TO NUMERIC block where the sequence is made bipolar. Then, the PN sequence is fed into a TAPPED DELAY LINE block where it is converted into vector format to allow for storage into the VECTOR RAM block. To control addressing of the VECTOR RAM, the VCC and ADDRESS COUNTER blocks are used.

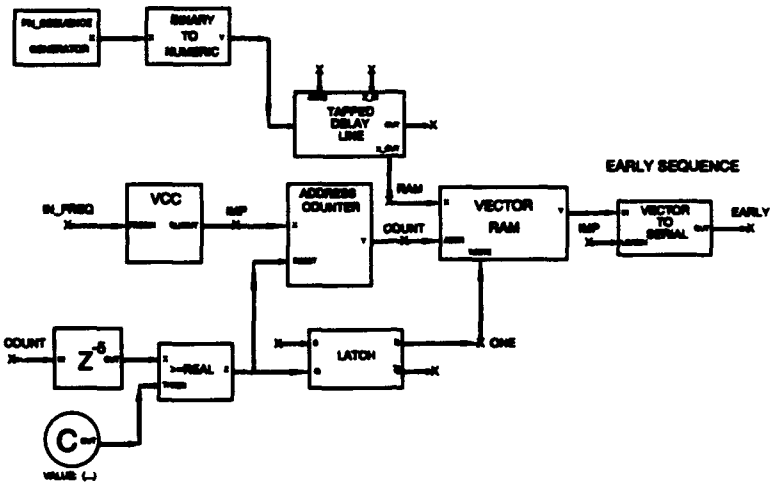


FIGURE 8 PN SOURCE BLOCK

The VCC block [Fig 9] is actually the Voltage Controlled Clock (VCC) and initially is used to store the PN sequence in the VECTOR RAM. A FUNCTION GENERATOR generates a square wave at a frequency of 20 Hz. The square wave is then converted into a series of impulses by the RISING EDGE TRUE block at a frequency of 20 Hz. The frequency of the impulses is twice the chip rate to allow for the sequence to be written into two consecutive places in memory in the VECTOR RAM. A timer block is ORed with the impulses so the first rising edge of the square wave is not converted into an impulse. This will allow address zero of the VECTOR RAM to be written into.

After the PN sequence is stored in the VECTOR RAM and the control switch of the DLL is turned on, the frequency from the loop filter passing through a GAIN SCALAR block, K_c, will control the VECTOR RAM addressing.

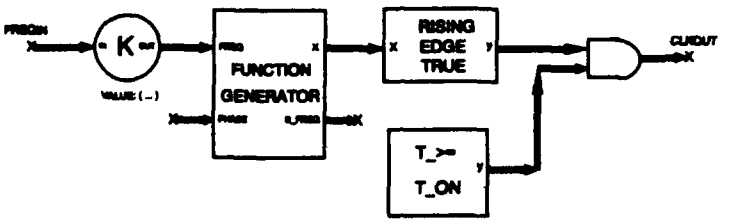


FIGURE 9 VCC BLOCK

The ADDRESS COUNTER block [Fig 10] contains a counter to actually control the RAM addressing. The impulses from the VCC block enter a >REAL block where they are compared with a threshold of 0.5. If the impulse is greater then 0.5, then a TRUE is output creating a square wave. The square wave is output through an invertor, passed through an OR gate, and fed into the "hold" input of a SIMPLE COUNTER block. When the hold on the SIMPLE COUNTER is pulled high by the inverted square wave, the RAM address is incremented by one. The counter is reset to one after the 61 addresses have been accessed. Even though the counter is not reset to zero, this has no effect on the simulation since addresses zero and one of the RAM contain the same stored chip information. The frequency of the square wave is 20 Hz.

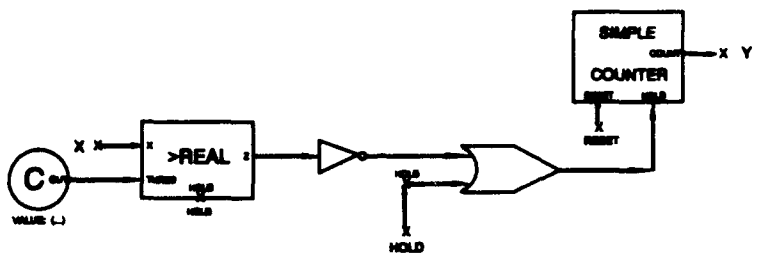


FIGURE 10 ADDRESS COUNTER BLOCK

After the VECTOR RAM is loaded with the PN sequence, the addressing frequency is the rest frequency plus the input frequency from the DLL. Also, the output of the SIMPLE COUNTER in the ADDRESS COUNTER block is passed through a DELAY block and into a >=REAL block where the count is compared to a threshold value of 61. This delay is required by the software since no feedback can occur without a delay in the loop. When the count reaches 61, a TRUE is output and the SIMPLE COUNTER in the ADDRESS COUNTER block is reset to one. The output of the >=REAL block is also sent to a LATCH block.

The LATCH block outputs a zero to the VECTOR RAM which enables the RAM to read and store the PN sequence. After the counter reaches 61, a TRUE signal will reset the LATCH and it will output a one for the rest of the simulation to enable the RAM to perform a read function. The output of the VECTOR RAM is passed through a VECTOR TO SERIAL block prior to the sequence being output of the PN SOURCE block.

The VECTOR TO SERIAL block converts the vector format of the PN sequence in the RAM to a serial format for output into the loop. The impulse train from the VCC block also controls the VECTOR TO SERIAL block. The vector input is latched in the block when "loadin" goes TRUE. If all the PN sequence is output and "loadin" goes FALSE, then it cycles back to the first element. This is the entire format to produce the "early" PN sequence.

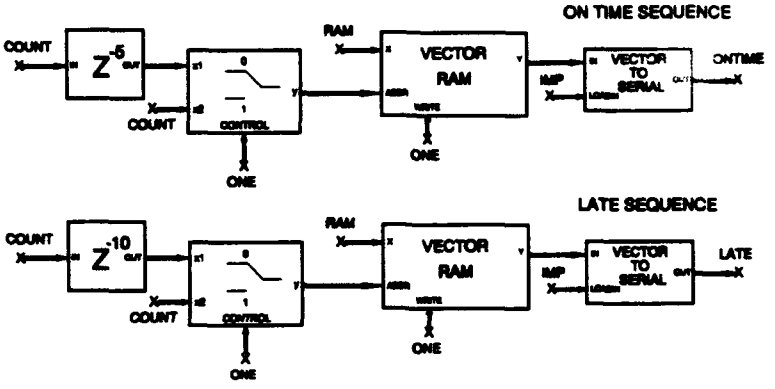


FIGURE 11 PN SOURCE BLOCK CONTINUED

To produce the "ontime" and "late" sequences, the count signal is passed through DELAY blocks of -5 for the ontime and -10 for the late versions and into switch blocks [Fig 11]. The switch blocks enable the two VECTOR RAMS to be written into and then switched through the DELAY blocks for the rest of the simulation to enable the RAMs to be read from. The VECTOR TO SERIAL blocks work the same as the one for the early sequence.

2.4.3 INTEGRATE & DUMP BLOCK. The INTEGRATOR & DUMP block is the standard block used in COMDISCO software. It is used in place of the INTEGRATOR block created by

Hird because it is more stable [HIR92]. It contains the backward difference method characterized as $y[n] = y[n-1] + x[n]/f_s$, where f_s is the sampling frequency. Hird's integrator contained the method $y[n] = y[n-1] + x[n]$. To fulfill the software requirement requiring a delay within a feedback loop, Hird took his output from the integrator at the $y[n-1]$ point. The clock rate controlling the integrate and dump function was changed from 140 to 10 intervals to allow the integration to be performed over one chip. Having the clock rate at 140 intervals equates to integrating over 14 chips and was a source of error in Hird's DLL.

Since the software requires a delay in a feedback loop, UNIT DELAY blocks are added at the output of the PN SOURCE block. A T_<= T_OFF block was added to initially reset the clock which compensates for the delay blocks contained within the loop. By resetting the clock after a certain number of intervals, any remaining delay error (prior to turning on the noise) was reduced to approximately zero.

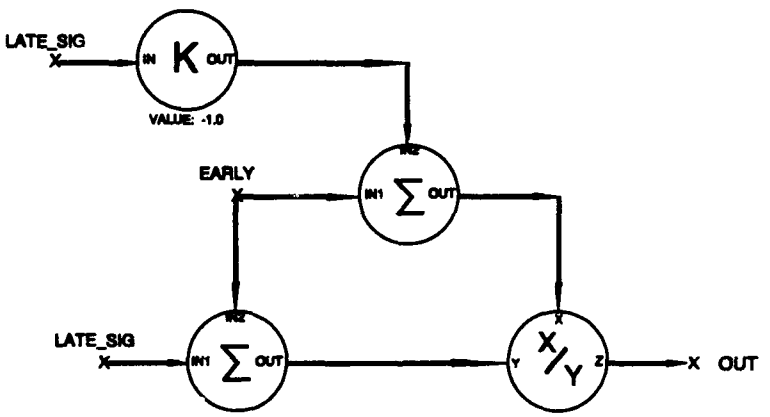


FIGURE 12 DELAY DETECTOR BLOCK

2.4.4 THE DELAY DETECTOR BLOCK. The configuration of the DELAY DETECTOR block is shown in Figure 12. The early signal is fed into two summers. One summer adds the early signal, E, to the late signal, L, and the other adds it to a negative version of the late signal. A negative version of the late signal is created by passing the late signal through a GAIN SCALER with a value of -1. A DIVISOR divides the difference of the early and late signals (E-L) by the sum of the two (E+L) to create an output of (E-L)/(E+L). If the denominator is zero, then the DIVISOR is set to output a zero and continue the simulation.

2.4.5 THE LOOP FILTER BLOCK. The components of the LOOP FILTER block are depicted in Figure 13. The Laplace Transform of the loop is $F(s) = G_1 + G_2/s$. When G_2 is set to zero, the order of the loop filter is zero and the order of the loop is first-order. When G_2 is set to any value other than zero, the filter becomes first-order and the loop becomes a second-order loop. In theory, a second-order loop can track out the effects of a frequency step (Doppler shift).

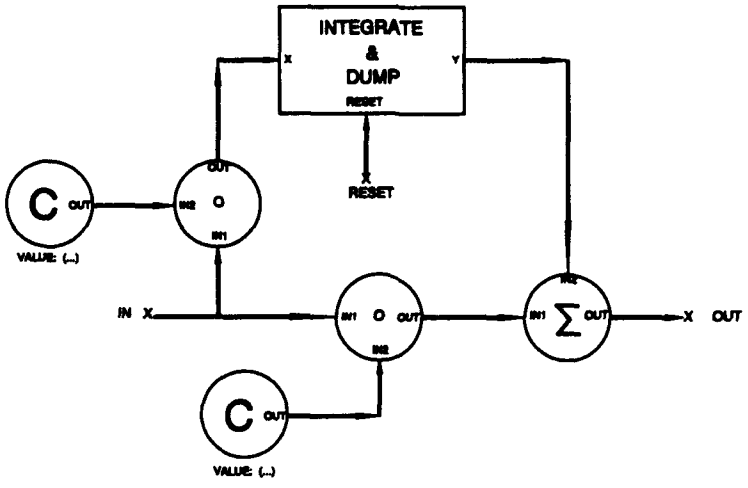


FIGURE 13 LOOP FILTER BLOCK

The input signal branches into two mixers where it is multiplied by loop gains of G_1 and G_2. The multiplication of the input signal with G_1 is passed through an INTEGRATE & DUMP block (Discussed in section 2.4.3) then summed with the multiplicative of the input signal and G_1. The resetting of the integrator is controlled by a clock with an interval setting of ten. This allows the signal to be integrated over one chip.

When Hird set the value of G_2 to anything other then zero, his DLL became unstable [HIR92]. It was determined that the clock signal was not entering the INTEGRATOR block in his LOOP FILTER, therefore, the integrator was constantly integrating. This also caused an error in his results.

When the clock signal was allowed to enter the LOOP FILTER block, the integrator performed an integrate and dump function over one chip. This allowed G_2 to be set to a value other than zero and increase the loop to second-order. Changing Hird's INTEGRATOR blocks to INTEGRATE & DUMP blocks increased the stability of the loop.

2.5 CHAPTER SUMMARY

This chapter described the equations developed for DLL analysis and discussed the software model used to simulate the DLL. Various differences between this DLL and the one presented by Hird [HIR92] were discussed. One difference was changing the clock rate to integrate over one chip. Anther difference was to introduce the clock rate into the Loop Filter, thereby, causing the loop filter integrator to also integrate over one chip. Finally, changing the INTEGRATOR blocks to INTEGRATE & DUMP blocks increased the stability of the loop.

III. DELAY-LOCK LOOP DEGRADED WITH AWGN

3.1 CHAPTER OVERVIEW

This chapter presents the equations used to analyze the DLL degraded only with AWGN, the methodology used to measure loop performance, and the results of the simulations. Section 3.2 describes the equations used to analyze the loop in the presence of AWGN. Section 3.3 introduces the methods used to determine loop performance in the presence of noise. Section 3.4 displays the results of the simulations for different values of N_0/P, G_2, and K_cG_1. Section 3.5 lists suggested operating points. Finally, Section 3.6 concludes the chapter with a summary.

3.2 DLL ANALYSIS IN THE PRESENCE OF AWGN

Let AWGN be added to the input signal which is expressed as [HIR92]:

$$s(t) = \sqrt{P}\, g(t-\tau)m(t-\tau)\cos[\omega_0 t + \Theta(t)] + n(t) \qquad (12)$$

where: P is the power in the input signal, s(t), in watts

τ is the transmission or propagation delay of the signal in seconds

$g(t-\tau)$ is a repeating pseudo-noise (PN) sequence with values ± 1, M chips in length, each chip of T sec duration. The period of $g(t-\tau)$ is MT seconds.

$m(t-\tau)$ is a message signal with values ± 1, edge synchronized with $g(t-\tau)$.

ω_0 is the nominal carrier frequency in rad/sec.

$\Theta(t) \triangleq \Delta\omega t + \theta_0$ is the channel rotation resulting from dynamics such as Doppler.

$\Delta\omega \triangleq [\omega - \omega_0]$ is the offset between the nominal carrier frequency, ω_0, and the actual frequency of s(t), ω.

n(t) is AWGN with a bandpass representation of [HIR92]:

$$n(t) = N_c(t)\cos[\omega_0 t + \Theta(t)] - N_s(t)\sin[\omega_0 t + \Theta(t)] \tag{13}$$

where $N_c(t)$ is that portion of the noise which is the in-phase component and $N_s(t)$ is the quadrature component. Both are independent and stationary processes with one-sided noise spectral density, N_0 watt-sec/rad.

(Refer to Figure 1 for the following analysis)

After mixing the input signal with the early PN sequence, the output of the mixer is defined as:

$$\begin{aligned} E(t) = {} & \sqrt{P}\,K_m\, m(t-\tau)\overline{g(t-\tau)g\left(t-\hat{\tau}+\frac{\Delta}{2}\right)}\cos[\omega_0 t + \Theta(t)] \\ & + \left\{\left[g(t-\tau)g\left(t-\hat{\tau}+\frac{\Delta}{2}\right) - \overline{g(t-\tau)g\left(t-\hat{\tau}+\frac{\Delta}{2}\right)}\right]\right. \\ & \left. \bullet\ \sqrt{P}\,K_m\, m(t-\tau)\cos[\omega_0 t + \Theta(t)]\right\} \\ & + K_m\, g\left(t-\hat{\tau}+\frac{\Delta}{2}\right) n(t) \end{aligned} \tag{14}$$

where K_m is the mixer gain and the overbar represents statistical expectation (correlation). The middle term is the self-noise term and can be neglected [SPI63].

After the BPF, the signal in the early arm of the loop becomes:

$$\tilde{E}(t) = \sqrt{P}K_m \tilde{m}(t-\tau)R_{g_+}(\tau_e/\Delta)\cos[\omega_0 t + \Theta(t)] + K_m \tilde{g}\left(t-\hat{\tau}+\frac{\Delta}{2}\right)\tilde{n}(t) \quad (15)$$

where ~ represents a filtered signal and $\tau_e \triangleq \tau - \hat{\tau}$.

Similarly, the signal from the late arm of the loop is:

$$\tilde{L}(t) = \sqrt{P}K_m \tilde{m}(t-\tau)R_{g_-}(\tau_e/\Delta)\cos[\omega_0 t + \Theta(t)] + K_m \tilde{g}\left(t-\hat{\tau}-\frac{\Delta}{2}\right)\tilde{n}(t) \quad (16)$$

The error signal output, e(t), from the delay error detector now becomes:

$$e(t) = \frac{\tilde{E}-\tilde{L}}{\tilde{E}+\tilde{L}} = \frac{A(t)D_g(\tau_e/\Delta) + n_-(t)}{A(t)S_g(\tau_e/\Delta) + n_+(t)} \quad (17)$$

where: $A(t) = \sqrt{P}K_m\tilde{m}(t-\tau)\cos[\omega_0 t + \theta(t)]$

$D_g(\tau_e/\Delta) = R_{g_+}(\tau_e/\Delta) - R_{g_-}(\tau_e/\Delta)$

$$S_g(\tau_e/\Delta) = R_{g+}(\tau_e/\Delta) + R_{g-}(\tau_e/\Delta)$$

$$n_\pm(t) = K_m \tilde{n}(t) [\tilde{g}(t - \hat{t} + \Delta/2) \pm \tilde{g}(t - \hat{t} - \Delta/2)]$$

The output of the VCC is defined as [GIL66]:

$$\frac{\hat{\tau}}{\Delta} = H(p)\left[\frac{A(t)\frac{\tau}{\Delta} + n_-(t)}{A(t) + n_+(t)}\right] \tag{18}$$

which is the linearized loop equation in the presence of noise with the restriction of $|\tau_e| \leq \Delta/2$ to cause the detector output to remain in the linear region.

Since the noise of the input signal is assumed white, the noise component of the error signal, $n_\pm(t)$, will also be white [COO86]. The power spectral density (PSD) of the noise component is represented by [HIR92]:

$$S_n(\omega) = 2\left(\frac{M+1}{M}\right)\frac{N_0}{2} \tag{19}$$

$$= N_0 \qquad \textit{for large } M$$

where 2[(M+1)/M] is the average power in the signal [g(t-$\hat{t}$-Δ/2) ± g(t-$\hat{t}$+Δ/2)] [LIN72].

The variance of the delay error, τ_{en}, due to noise then becomes [GIL66]:

$$\left(\frac{\sigma_{en}}{\Delta}\right)^2 = \frac{1}{P}\int_{-\infty}^{\infty} S_n(\omega)|H(\omega)|^2 d\omega$$

$$= \frac{N_0 W_L}{P} \qquad (20)$$

$$= \frac{N_0}{P}\left(K_c G_1 + \frac{G_2}{2G_1}\right)$$

where W_L is the two-sided loop bandwidth and σ^2_{en} is the variance of AWGN. K_c is the gain of the VCC, and G_1 and G_2 are loop filter gains.

The probability of losing lock is [COO86]:

$$P\left[\tau_{en} > \frac{\Delta}{2}\right] = 2Q\left[\frac{\Delta}{2\sigma_{en}}\right]$$

$$= 2Q\left[\frac{1}{2}\sqrt{\frac{P}{N_0 W_L}}\right] \qquad (21)$$

The SNR of the loop signal is the term under the square root sign (SNR = P/N_0W_L) where N_0W_L is the noise power, and $N_0 = \sigma^2_{en}/2$. The power in the input signal, P, is the square of the peak amplitude of the signal.

The loop will maintain lock as long as $|\tau_{en}| \leq \Delta/2$, therefore, the normalized delay error becomes [HIR92]:

$$\frac{T}{2} \geq T\sqrt{\frac{N_0}{P}\left(K_c G_1 + \frac{G_2}{2G_1}\right)} \tag{22}$$

To define loop performance corrupted only by AWGN, equation (22) becomes:

$$1 \geq 2\sqrt{\frac{N_0}{P}\left(K_c G_1 + \frac{G_2}{2G_1}\right)} \tag{23}$$

which represents the delay error ceiling. A delay error greater than the delay error ceiling corresponds to a cycle slip. A cycle slip occurs when the local PN sequence has slipped forward or aft of the repeating input PN sequence and locked onto another element of it. By setting G_2 and N_0/P to predicted levels, $K_c G_1$ can be varied to determine optimum loop performance.

3.3 DLL METHODOLOGY

In general, the DLL performed better than theoretical predictions indicated. The loop tracked a signal degraded only with AWGN. By setting the parameters G_2, K_c, and N_0/P to predicted values, G_1 was varied to obtain simulation results. P is the power of the input signal which is the square of the peak amplitude of the signal (which is one in this case). N_0 is the one-sided PSD of the noise and was set by changing the variance of the noise and multiplying by two.

Multiplying by two is necessary since Gaussian noise has a two-sided PSD. K_c is the gain of the VCC and G_2 is the gain of the loop filter that determines the amount of 'memory' the loop filter contains. G_1 is also a loop filter gain. Simulations were run varying G_1. A typical value of N_0/P for a GPS signal is 1.4×10^{-4} sec/rad [SPI80].

The maximum loop delay error was obtained by overlaying the input PN sequence (prior to the NOISE MAKER block) with the early PN sequence from the PN SOURCE block and visually counting the number of samples between zero crossings. By dividing the maximum number of samples of delay by the total number of samples in each chip, at that moment in time, the normalized delay error is obtained. In this case, divide the maximum error samples found by 100 samples/chip.

Each simulation data point was created by performing multiple simulations in which only the noise seed was varied. By dividing the total maximum delay errors found in each simulation by the number of simulations performed, an average value of each point was obtained for plotting purposes. Each simulation was performed over an interval of 20,000 iterations. The stars and diamonds represent simulation points on the plots.

3.4 Simulation Results

Figure 14 displays the results obtained by setting $G_2 = 0$ to emulate a first-order loop. The straight slanted lines indicate theoretical loop performance for different values of N_0/P. The loop performed better than predicted and the increased delay error for increased K_cG_1 is apparent. Also, the trend of increased delay error with increased N_0/P is readily apparent. Increasing K_cG_1 greater than a value of five results in the loop becoming unstable and breaking the delay error ceiling. The exact reason the threshold effect occurred is unknown.

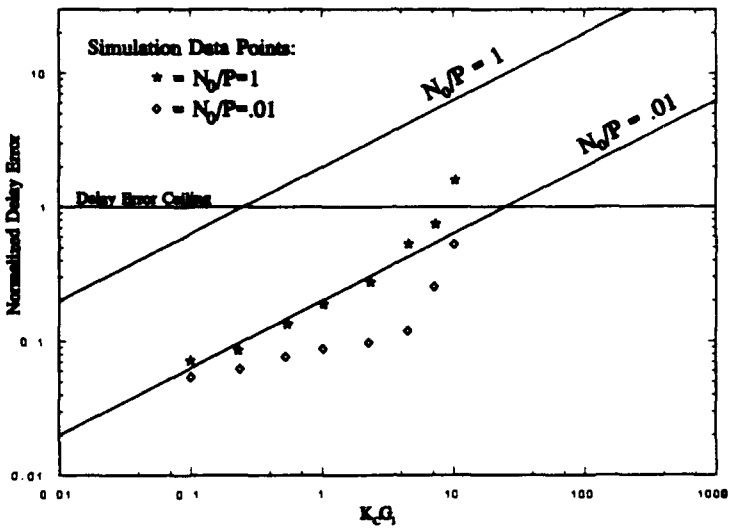


FIGURE 14 DLL SIMULATION RESULTS WITH $G_2 = 0$.

The result of setting $G_2 = 1$ and plotting the simulations is depicted in Figure 15. As shown, the second-order loop performed equally well for both values of N_0/P. The trend of the simulation points follows the shape of the theoretical curves. Values of K_cG_1 greater than ten resulted in the loop becoming unstable and a threshold effect occurring which was not predicted by theory. The threshold effect may be due to the loop reaching a feedback instability point where errors are compounded causing the loop to drift further from lock.

Figure 16 is the result of setting $G_2 = 2$. The trend of the simulation data points generally follows the theoretical curves. Also, both values of N_0/P are less than predicted by theory. The loop still became unstable for values of K_cG_1 greater than ten.

By setting $G_2 = 3$, Figure 17 demonstrates that higher values of G_2 result in the loop becoming unstable sooner. However, the trend of the simulation points follow the theoretical curves up to the threshold region. Requiring a delay within any feedback loop, or the sensitivity of the loop to different values of N_0/P could have resulted in decreased loop performance.

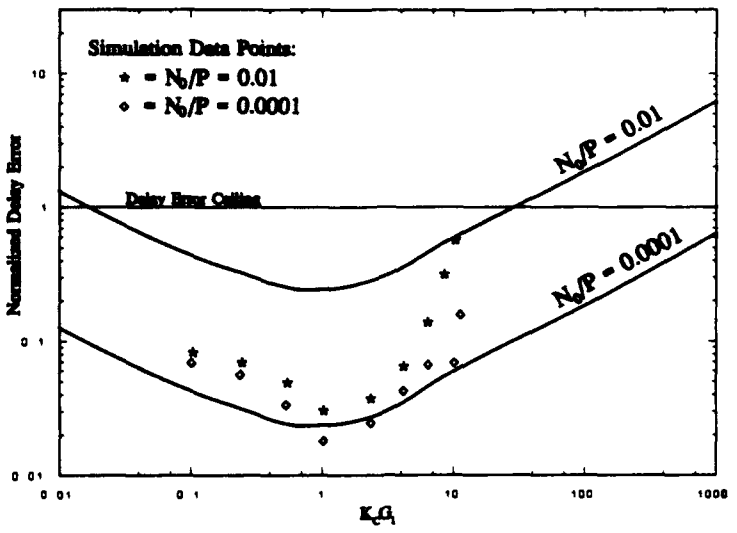


FIGURE 15 DLL SIMULATION RESULTS WITH G_2 = 1

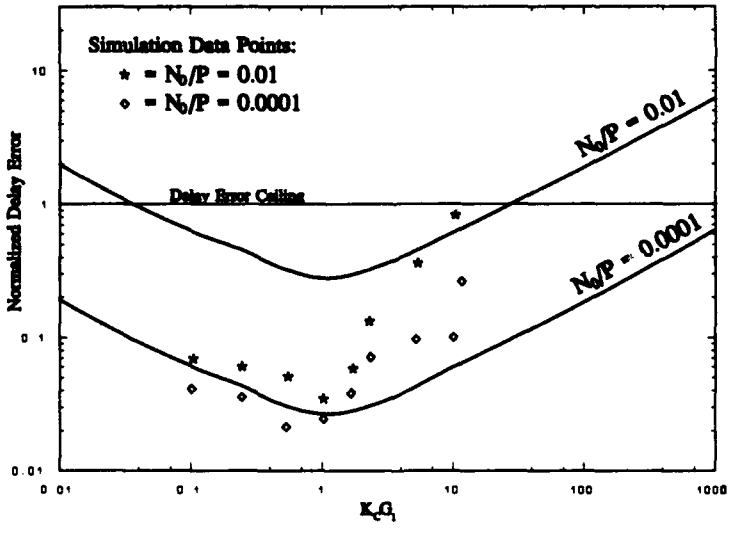


FIGURE 16 DLL SIMULATION RESULTS WITH G_2= 2

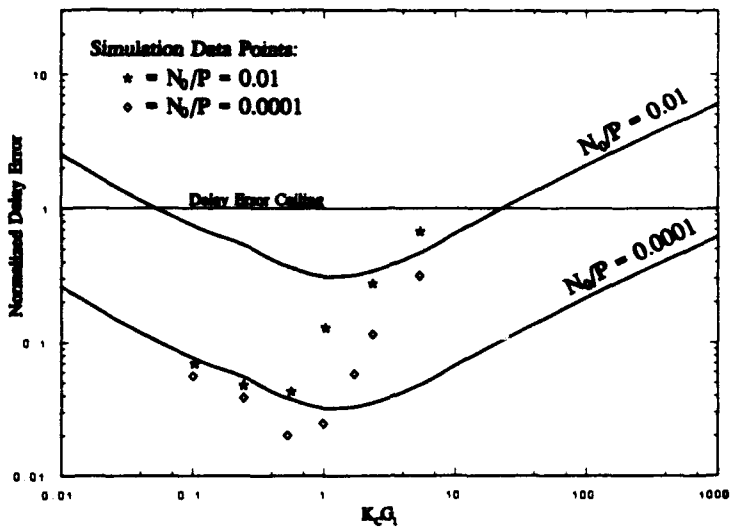


FIGURE 17 DLL SIMULATION RESULTS WITH G_2= 3

3.5 SUGGESTED OPERATING POINTS

For the DLL using second-order loops, operating points for maximum loop performance, in the presence of AWGN, can be determined from the plots. The loop appears to operate best at $G_2 = 1$ and $K_cG_1 = 1$ where the maximum delay error is approximately 0.02 for $N_0/P = 0.0001$ and 0.03 for $N_0/P = 0.01$. Another good operating point is $G_2 = 2$ and $K_cG_1 = 0.5$.

3.6 CHAPTER SUMMARY

The equations analyzing the DLL corrupted by AWGN were discussed along with the method used to perform the simulations. The theoretical delay error ceiling and maximum delay curves were derived and plotted on each graph. Loop performance generally follows along theoretical lines. A threshold effect occurred at certain values of K_cG_1 which limited the range of the simulations. Values of K_cG_1 greater than the threshold region resulted in loop instability in which cycle slips occurred. As G_2 was increased, the value of K_cG_1 where the threshold effect occurred decreased which allowed the loop to become unstable for smaller values of K_cG_1.

IV. MODIFIED TANLOCK LOOP SIMULATION

4.1 CHAPTER OVERVIEW

This chapter presents an overview and analysis of a digital Modified Tanlock Loop (MTLL) model. Section 4.2 presents a basic overview of a MTLL. In Section 4.3 the equations developed to analyze a MTLL are described. Section 4.4 introduces the MTLL model and its individual components used to simulate a MTLL. Finally, Section 4.5 summarizes the chapter.

4.2 MODIFIED TANLOCK LOOP OVERVIEW

A phase-locked loop (PLL) is used to achieve phase synchronization with an incoming carrier signal. This PLL (called a carrier tracking loop) generates a local reference signal that is controlled by a Voltage Controlled Oscillator (VCO) and is mixed with the incoming signal to achieve carrier synchronization. The type of carrier tracking loop being simulated in this thesis is called a digital Modified Tanlock Loop (MTLL) [Fig 18].

The MTLL is a non-uniform sampling digital PLL whose main feature is the inverse tangent ($\tan^{-1}$) phase detector [LEE82]. The MTLL can be characterized by a linear difference equation which provides many advantages over other conventional PLLs. First, the MTLL has a wider lock range and less steady-state phase error for a first-order loop with an input having a frequency offset. Second, the MTLL can be locked independently of initial phase error for the same region of parameter values. Finally, enhanced noise immunity and insensitivity to variations of input signal power are achieved.

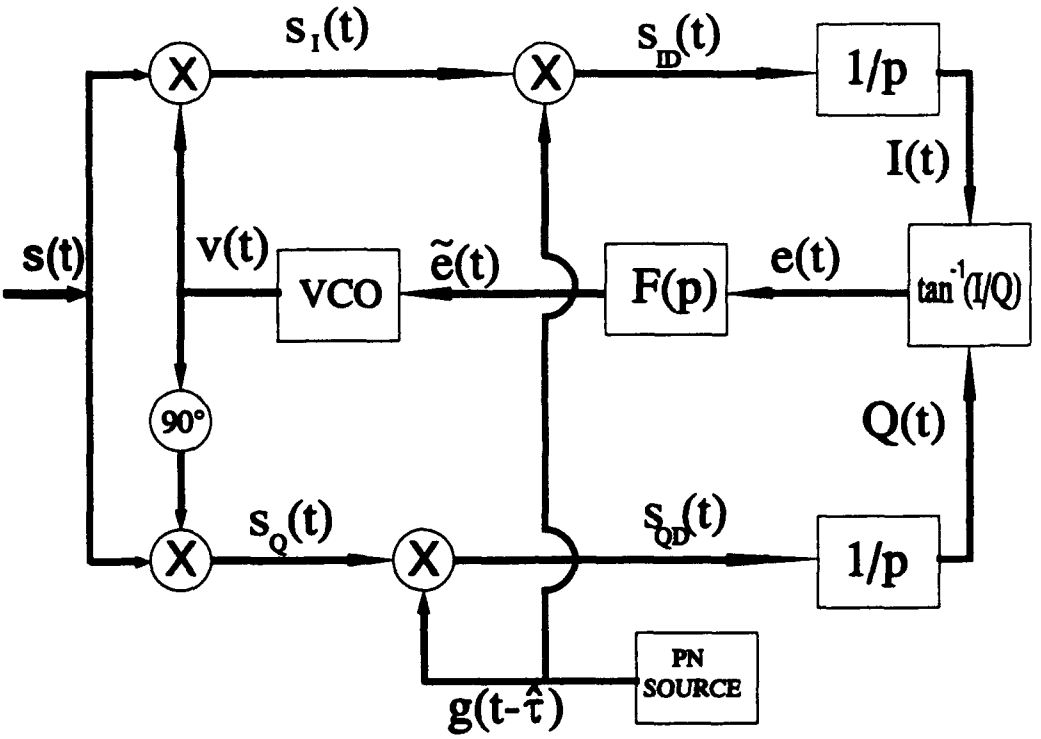


FIGURE 18 MODIFIED TANLOCK LOOP

The signal output from the VCO, v(t), is split into in-phase and 90° out of phase components. These components are mixed with the input signal, s(t), to create in-phase, $s_I(t)$, and quadrature, $s_Q(t)$, signals. Then, these signals are mixed with a locally generated PN source, $g(t - \hat{\tau})$, which despreads the signals. The despread signals, $s_{ID}(t)$ and $s_{QD}(t)$, are then passed through integrators, 1/p, to create inphase, I(t), and quadrature, Q(t), arms of the loop respectively. The phase detector finds the arctangent of I(t) over Q(t) to create a phase error signal, e(t). The phase error signal is passed through a loop filter, F(p), and the corresponding filtered error signal, $\tilde{e}(t)$, is used to control the VCO by varying the output frequency and phase.

4.3 MODIFIED TANLOCK LOOP ANALYSIS

The input signal, s(t), contains a data signal, m(t), and a spreading signal, g(t), BPSK modulated onto a carrier signal of frequency ω_0. In the absence of noise, the input signal is written as [HIR92]:

$$s(t) = \sqrt{2P}\, m(t)\, g(t) \sin[\omega_0 t + \Theta(t)] \tag{24}$$

where: P is the power of the received signal, s(t), in watts

m(t) is the unknown message signal BPSK modulated on the carrier, having values of ± 1

g(t) is the pseudo-random noise (PN) spreading signal, edge synchronized with m(t), having values of ± 1

$\Theta(t) \triangleq (\Delta\omega t + \theta_0)$ and is the result of channel rotation

$\Delta\omega \triangleq (\omega - \omega_0)$ rad/sec

ω is the actual frequency of s(t) in radians/sec

θ_0 is an arbitrary initial phase offset, in radians

The output of the VCO, v(t), is defined as [HIR92]:

$$v(t) = \sqrt{2P_v}\, \cos[\omega_0 t + \Theta_v(t)] \tag{25}$$

where: P_v is the power of the VCO output, v(t), in watts

$\Theta_v(t) = (\Delta\omega_v t + \theta_v)$ radians

$\Delta\omega_v = (\omega_v - \omega_0)$ rad/sec

ω_v is the actual frequency of the VCO output, in radians/sec

θ_v is an arbitrary phase offset, in radians

After mixing s(t) with v(t), the signal after the first mixer becomes:

$$s_I(t) = \sqrt{PP_v}\, K_m\, m(t) g(t) \{\sin[2\omega_0 t + \Theta(t) + \Theta_v(t)] + \sin[\Theta(t) - \Theta_v(t)]\} \qquad (26)$$

where K_m is the gain of the first mixer.

After mixing the PN spreading signal, $g(t - \hat{t})$, with $s_I(t)$ and passing the result through a BPF, 1/p, the inphase signal, I(t), becomes:

$$I(t) = \sqrt{PP_v}\, K_m K_g \tilde{m}(t) R_g(\tau_e) \sin[\Theta_e(t)] \qquad (27)$$

where K_g is the gain of the second mixer, $\Theta_e(t) = [\Theta(t) - \Theta_v(t)]$ which is the phase error between the VCO output and the received signal, and ~ represents a signal that has been filtered.

Similarly, an analysis of the other loop arm generates a quadrature signal, Q(t), which is defined as:

$$Q(t) = \sqrt{PP_v}K_m K_g \tilde{m}(t) R_g(\tau_e)\cos[\Theta_e(t)] \tag{28}$$

The phase detector output, e(t), is generated by $e(t) = \tan^{-1}[I(t)/Q(t)]$ which becomes:

$$e(t) = \tan^{-1}\left[\frac{\sqrt{P}\,\tilde{m}(t)R_g(\tau_e)\sin[\Theta_e(t)]}{\sqrt{P}\,\tilde{m}(t)R_g(\tau_e)\cos[\Theta_e(t)]}\right] \tag{29}$$

which is linear over the region $-\pi \le \Theta_e \le \pi$ [LEE82]. Note the phase error simplifies to $\Theta_e(t)$ in the absence of noise. Mixer gains, K_m and K_g, and the VCO output power, P_v, have been eliminated from the equations, hence, no automatic gain control circuitry is required to control loop arm imbalance.

The VCO acts as an integrator, therefore, the input to the VCO is the derivative of the desired phase component. This observation leads to the stochastic integro-differential equation describing the loop operation [HIR92]:

$$\begin{aligned}\Theta_v(t) &= \int K_0 \tilde{e}(t)\,dt \\ &= \frac{K_0 F(p)\,e(t)}{p}\end{aligned} \tag{30}$$

where F(p) is the loop filter, G_1 and G_2 are loop gain parameters, and K_0 is the VCO gain constant. The loop filter transfer function in the Laplace domain is $F(s) = G_1 + G_2/s$ where the Laplace Transform variable $s = j\omega = p$.

The loop bandwidth, W_L, is defined as [COO86]:

$$W_L = \int_{-\infty}^{\infty} |H(\omega)|^2 \, d\omega = \frac{1}{2}\left(K_0 G_1 + \frac{G_2}{G_1}\right) \frac{rads}{sec} \tag{31}$$

4.4 Modified Tanlock Loop Model

The MTLL model used to perform the simulations is depicted in Figure 19. The MTLL INPUT SIGNAL block generates a carrier signal (sine wave) which is passed through a UNIT DELAY block. Each arm of the loop contains a UNIT DELAY block to meet the software requirement of a delay required in a feedback loop. Therefore, the signal input must pass through a UNIT DELAY block to compensate for the loop delays. This helped to reduce an initial phase offset, which occurred prior to any noise being added to the loop, from 18 degrees to approximately 3 degrees. Hird's MTLL did not contain any input signal delay, therefore, the initial phase offset in his loop probably contributed to his simulation errors.

Next, the input sine wave is split and sent to two mixers. The upper mixer multiplies the input signal with the REAL (cosine) part of the VCO output signal to create a sine signal which is now the in-phase arm of the loop. The lower mixer multiplies the input signal with the

imaginary (sine) part of the VCO output signal to create a 90 degree out-of-phase (cosine) signal which is now the quadrature arm of the loop.

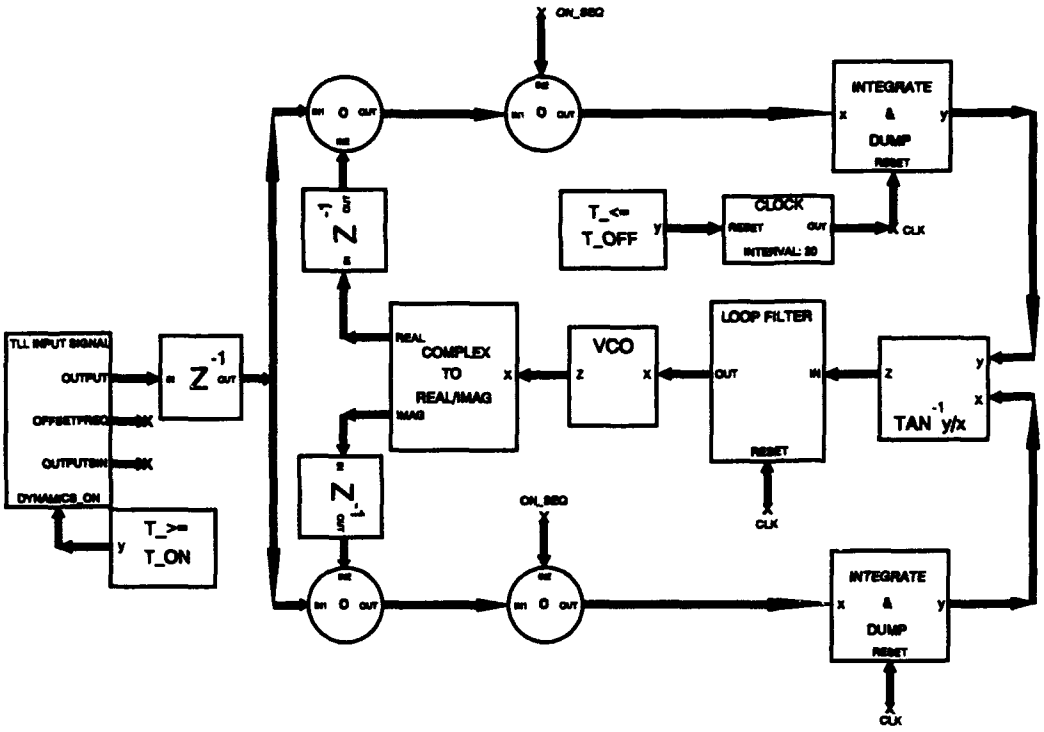


FIGURE 19 MODIFIED TANLOCK LOOP MODEL

Then, the two arms are mixed with the "on_seq" signal which equates to the "on-time" signal from the DLL. For the purposes of testing, the ontime signal from the DLL was set to a constant value of one. Next, the two arms are fed into INTEGRATE & DUMP blocks. The integrate and dump cycles are controlled by a clock. Resetting the clock to zero after the first interval will reduce the remaining three degrees phase offset to approximately zero by compensating for the required delay blocks within the loop.

The in-phase, I, and quadrature, Q, arms of the loop are then fed into an arctangent block, $\tan^{-1}(y/x)$, which produces a phase error signal. Next, the error signal is fed into a LOOP FILTER block which filters the error signal and passes the result to the VCO. The VCO adjusts its frequency and phase according to the filtered error signal. The VCO uses the Simpson method of integration (described in Comdisco Systems software manuals) where the output frequency is equal to the natural frequency, f_c (a parameter), plus the input voltage times the constant, K_0 (i.e. $f = f_c + vK_0$). Finally, the VCO output is fed into a COMPLEX TO REAL/IMAG block where its split into REAL (cosine) and IMAG (sine) signals. Now, each custom designed block by Hird will be examined in detail.

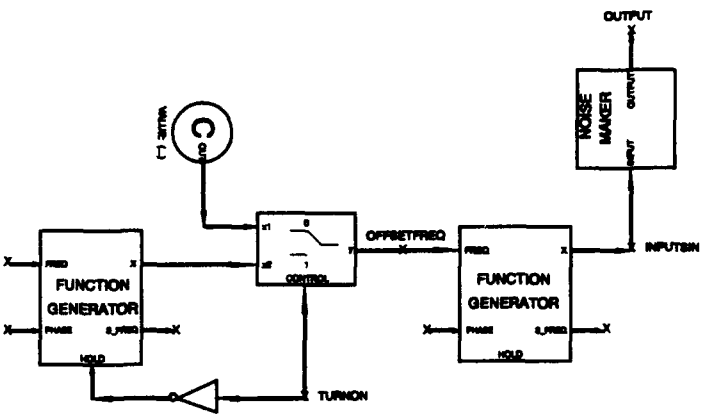


FIGURE 20 MTLL INPUT SIGNAL BLOCK

4.4.1 MTLL INPUT SIGNAL BLOCK. The components of the MTLL INPUT SIGNAL block are depicted in Figure 20. This block generates the sine signal used to simulate a carrier signal. Also, frequency steps and ramps (Doppler shifts) can be added from here. The first FUNCTION

GENERATOR block generates a triangle wave at a frequency of 25 Hz. It is used to add frequency steps and ramps to the input sinusoid. The triangle wave passes through a switch block and into another FUNCTION GENERATOR. The switch block is used to turn on/off the generation of the frequency steps and ramps used to corrupt the generation of the input sine wave.

The second FUNCTION GENERATOR generates the input sine wave at a frequency of 25 Hz and an amplitude of +/- 1 volt. The sine wave passes through a NOISE MAKER block (Described in Section 2.4.1, Figure 7) where AWGN is added. The nominal GPS carrier frequency is $2\pi 1.5$ GHz, however, for the ease of simulation, the carrier frequency is set to 25 Hz.

4.4.2 INTEGRATE & DUMP BLOCK. The INTEGRATE & DUMP block is the standard block used in COMDISCO software. It is used in place of the INTEGRATOR block created by Hird because it is more stable [HIR92]. It contains the backward difference method characterized as $y[n] = y[n-1] + x[n]/f_s$, where f_s is the sampling frequency. Hird's integrator contained the method $y[n] = y[n-1] + x[n]$. To fulfill the software requirement requiring a delay within a feedback loop, Hird took his output from the integrator at the y[n-1] point. The clock rate controlling the integrate and dump function was changed from 100 to 20 intervals to allow the integrator to integrate over one period. Having the clock rate at 100 intervals equates to integrating over 5 periods and could have been a source of error in Hird's MTLL.

Since the software requires a delay within a feedback loop, the UNIT DELAY blocks are added at the output of the COMPLEX TO REAL/IMAG block. A T_<= T_OFF block was added to initially reset the clock which compensates for the delay blocks contained within the loop. By resetting the clock after the first interval, the remaining 3 degree phase offset error (prior to turning on the noise) was reduced to approximately zero.

4.4.3 LOOP FILTER BLOCK. The components in the LOOP FILTER block (Discussed in Section 2.4.5) are shown in Figure 13. Again, the clock signal was not being fed into the LOOP FILTER block, therefore, the integrator was constantly integrating. This was probably another source of error and the reason Hird could not upgrade his loop from first to second-order.

4.5 CHAPTER SUMMARY

A MTLL overview discussed the basic block diagram being modeled in this simulation. Next, a MTLL analysis described the equations used to develop the model. Simulations of the loop model and all its individual components were discussed in detail. It was determined that Capt Hird's model failed to compensate for an initial phase offset. By adding an input signal delay and a timer to reset the clock controlling integration cycles, the initial phase offset was reduced to approximately zero. Also, the failure of the clock signal to reset the integrator within the loop filter was discovered in Hird's model, which prevented the order of the loop to be increased to second-order. Finally, the INTEGRATOR blocks within the loop were changed to INTEGRATE & DUMP blocks to increase loop stability.

V. *Modified Tanlock Loop Degraded With AWGN*

5.1 Chapter Overview

This chapter presents the equations used to analyze the MTLL corrupted only with AWGN, the methodology used to measure loop performance, and the results of the simulations. Section 5.2 describes the equations used to analyze the loop in the presence of AWGN. In Section 5.3, the method used to determine loop performance in the presence of noise is introduced. Section 5.4 displays the results of the simulations for different values of N_0/P, G_2, and K_0G_1. Section 5.5 lists suggested operating points. Finally, Section 5.6 summarizes the chapter.

5.2 MTLL Analysis In The Presence Of AWGN

Let AWGN be added to the input signal which is then expressed as [HIR92]:

$$s(t) = \sqrt{2P}\, m(t)\, g(t) \sin[\omega_0 t + \Theta(t)] + n(t) \tag{32}$$

where: P is the power of the received signal, s(t), in watts

m(t) is the unknown message signal BPSK modulated on the carrier, having values of ± 1

g(t) is the pseudo-random noise (PN) spreading signal, edge synchronized with m(t), having values of ± 1

$\Theta(t) \triangleq (\Delta\omega t + \theta_0)$ and is the result of channel rotation

$\Delta\omega \triangleq (\omega - \omega_0)$ rad/sec

ω is the actual frequency of s(t) in radians/sec

θ_0 is an arbitrary initial phase offset, in radians

n(t) is AWGN with a bandpass representation of [HIR92]:

$$n(t) = \sqrt{2}\left\{N_c(t)\cos[\omega_0 t + \Theta(t)] - N_s(t)\sin[\omega_0 t + \Theta(t)]\right\} \tag{33}$$

where $N_c(t)$ is that portion of the noise which is the in-phase component and $N_s(t)$ is the quadrature component. Both are independent stationary processes with single-sided PSD, N_0 watt-rad/sec.

From Figure 18, after mixing the input signal with the output of the VCO, v(t) (Equation 25), the top or in-phase arm of the loop becomes:

$$\begin{aligned} s_I(t) = {} & \sqrt{PP_v}\,K_m\,m(t)g(t)\left\{\sin[2\omega_0 t + \Theta(t) + \Theta_v(t)] + \sin[\Theta(t) - \Theta_v(t)]\right\} \\ & + \sqrt{2P_v}\cos[\omega_0 t + \Theta_v(t)]\,n(t) \end{aligned} \tag{34}$$

where k_m is the mixer gain. Then, the signal passes through a second mixer where the signal is multiplied with a despreading signal, g(t - $\hat{\tau}$).

After the second mixer and the BPF the in-phase arm of the loop is written as:

$$\begin{aligned} I(t) = {} & \sqrt{PP_v}K_m K_g \tilde{m}(t) R_g(\tau_e)\sin[\Theta_e(t)] \\ & + \sqrt{2P_v}\,K_m K_g\,\tilde{g}(t-\tau_e)\,\tilde{n}_e(t) \end{aligned} \tag{35}$$

where K_g is the gain of the second mixer and ~ represents a signal that has been filtered. The self-noise term from the second mixer can be neglected since loop bandwidth is much less than the PN sequence chip rate [SPI63]. Note $\Theta_e(t)$ is the phase error between the VCO output and the input signal and is defined as $[\Theta(t) - \Theta_v(t)]$.

The phase error noise term, $n_e(t)$, is defined as [HIR92]:

$$\tilde{n}_e(t) = \tilde{N}_c(t)\cos[\Theta_e(t)] - \tilde{N}_s(t)\sin[\Theta_e(t)] \tag{36}$$

Similarly, an analysis of the lower arm yields a quadrature signal of:

$$\begin{aligned} Q(t) = {} & \sqrt{PP_v}K_m K_g \tilde{m}(t)R_g(\tau_e)\cos[\Theta_e(t)] \\ & + \sqrt{2P_v}K_m K_g \tilde{g}(t-\tau_e)\tilde{n}'_e(t) \end{aligned} \tag{37}$$

where $n'_e(t)$ is a 90° phase shifted version of $n_e(t)$.

The output of the phase error detector, $e(t) = \tan^{-1}[y/x]$ then becomes:

$$e(t) = \tan^{-1}\left[\frac{\sqrt{P}\,\tilde{m}(t)R_g(\tau_e)\sin[\Theta_e(t)] + \sqrt{2}\,\tilde{g}(t-\tau_e)\tilde{n}_e(t)}{\sqrt{P}\,\tilde{m}(t)R_g(\tau_e)\cos[\Theta_e(t)] + \sqrt{2}\,\tilde{g}(t-\tau_e)\tilde{n}'_e(t)}\right] \tag{38}$$

which is the modification proposed by Lee and Un [LEE82]. Note mixer gains K_m and K_g, and the VCO output power, P_v, have been eliminated from the equations. It is exactly linear over the region $-\pi \le \Theta_e \le \pi$.

The variance of the phase error due to noise is [HIR92]:

$$\begin{aligned}\sigma^2_{en} &= \frac{1}{P}\int_{-\infty}^{\infty} S_n(\omega)\,|H(\omega)|^2\,d\omega \\ &= \frac{N_0 W_L}{2P} \\ &= \frac{N_0}{4P}\left(K_0 G_1 + \frac{G_2}{G_1}\right)\end{aligned} \tag{39}$$

where K_0 is the VCO gain, G_1 and G_2 are loop gains, the noise PSD $S_n(\omega) = N_0/2$, $H(\omega)$ is the loop transfer function, and W_L is the loop bandwidth. Note the SNR of the loop is the reciprocal of the phase error variance.

The loop will lose lock when the phase error exceeds π. Assuming Gaussian noise, the probability of the loop losing lock due to noise is then [COO86:351]:

$$\begin{aligned}P[\Theta_{en} > \pi] &= 2Q\left[\frac{\pi}{\sigma_{en}}\right] \\ &= 2Q\left[\pi\sqrt{\frac{2P}{N_0 W_L}}\right] \\ &= 2Q\left[\pi\sqrt{\frac{4P G_1}{N_0(K_0 G_1^2 + G_2)}}\right]\end{aligned} \tag{40}$$

Finally, for the loop to maintain lock on the input sinusoid the following restriction must apply [HIR92]:

$$\pi \geq \sqrt{\frac{N_0}{4P}\left(K_0 G_1 + \frac{G_2}{G_1}\right)} \tag{41}$$

To define loop performance corrupted only by AWGN, equation (41) becomes:

$$1 \geq \frac{1}{\pi}\sqrt{\frac{N_0}{4P}\left(K_0 G_1 + \frac{G_2}{G_1}\right)} \tag{42}$$

which represents the phase error ceiling. When the errors exceed the phase error ceiling, the phase error is greater then 180 degrees which leads to a cycle slip. A cycle slip occurs when the VCO output sinusoid leads or lags the input sinusoid by over 180 degrees. This tends to cause a VCO output sinusoidal period to more closely synchronize with a forward or aft version of the input signal's repeating sinusoidal periods.

5.3 MTLL METHODOLOGY

In general, the loop performed better than predicted by theory, however, a threshold effect limited the values of K_0G_1. By setting G_2 to any value other than zero, the order of the loop was increased from first to second-order. Simulations were performed by setting N_0/P, K_0, and G_2 to predicted values, then varying the G_1 parameter. The simulations were performed over 5000 iterations after which the VCO's output (IMAG arm of the COMPLEX TO IMAG/REAL block)

was superimposed on the input sinusoid. Visual inspection determined the number of samples between zero crossings. Dividing the magnitude of the number of samples by half a sinusoidal cycle produces a normalized phase error. For example, dividing the maximum number of phase errors found, before or after the input sinusoid, by half a sinusoidal cycle will produce the normalized phase error. In this case, divide the max error samples found by 10. Note, the carrier frequency used in the simulation is much lower than actual frequency.

Each simulation data point was plotted only after varying the noise seed then averaging the results together to form one simulation data point. For each simulation, the MTLL was allowed to lock onto the signal prior to any noise being added. The results of the simulations, where the solid lines represent the maximum phase error predicted by theory, are discussed next.

5.4 Simulation Results

Figure 21 depicts the performance of the loop with $G_2 = 0$ to emulate a first-order loop. The loop performed better than theory up to a value of $K_0G_1 = 20$. After that value, a threshold effect occurred not predicted by theory. The threshold effect may be due to the loop reaching a feedback instability point where errors are compounded causing the loop to drift further from lock. The loop quickly became unstable and difficult to measure in the region of the threshold effect. The trend of increased phase error with increased gain and increased N_0/P is evident.

By setting $G_2 = 1$, the loop is now a second-order loop and Figure 22 displays the results of the simulations. Figure 23 depicts the performance of the loop with $G_2 = 10$. As shown, the performance of the loop is better than predicted by theory until the value of K_0G_1 reaches the threshold point. The trend of increased phase error with increased gain is evident, as well as, the trend of increased phase error with increased N_0/P.

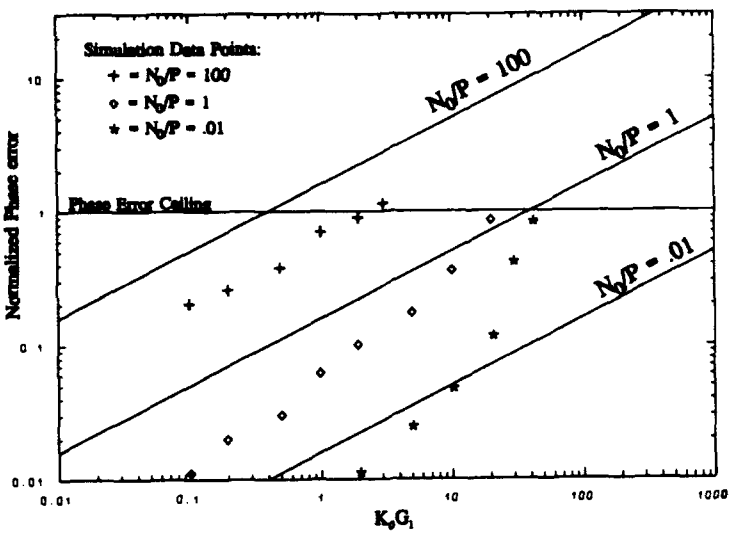


FIGURE 21 MTLL SIMULATION RESULTS WITH $G_2 = 0$

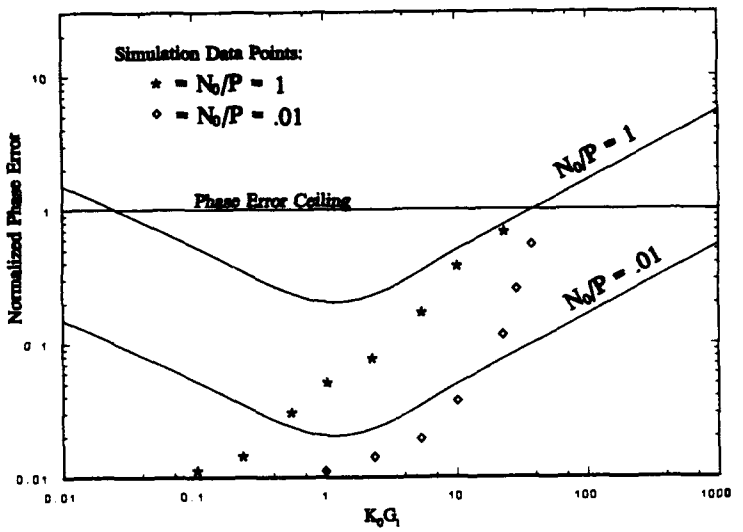


FIGURE 22 MTLL SIMULATION RESULTS WITH $G_2 = 1$

Figure 24 represents $G_2 = 40$ and Figure 25 displays the simulation results of $G_2 = 50$. There is not much difference between the two figures. They both represent the loop performing

better than theory until the K_0G_1 value reaches the threshold point. The trend of increased phase error points tends to follow the theoretical curves until the threshold point.

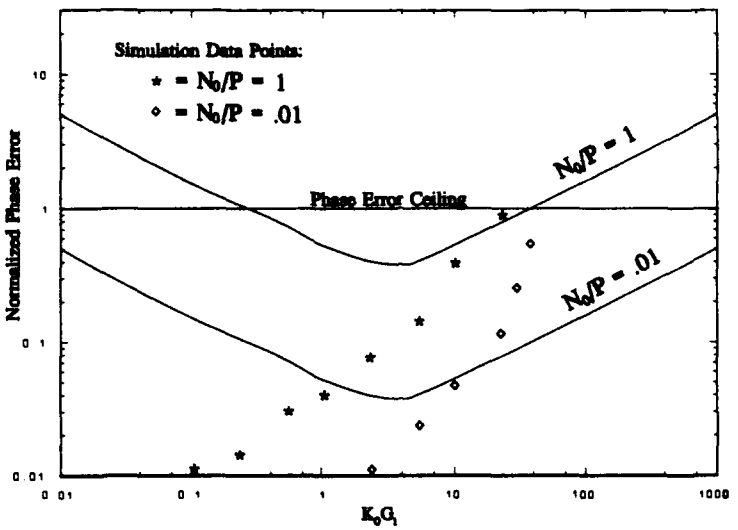


FIGURE 23 MTLL SIMULATION RESULTS WITH G_2 = 10

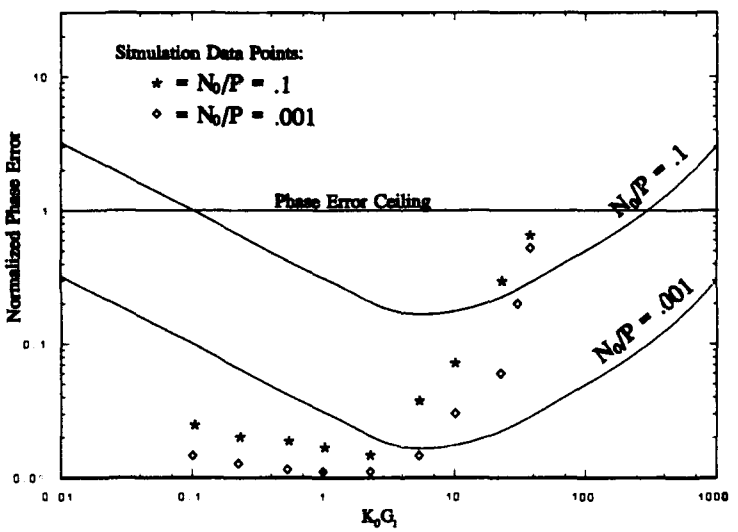


FIGURE 24 MTLL SIMULATION RESULTS WITH G_2 = 40

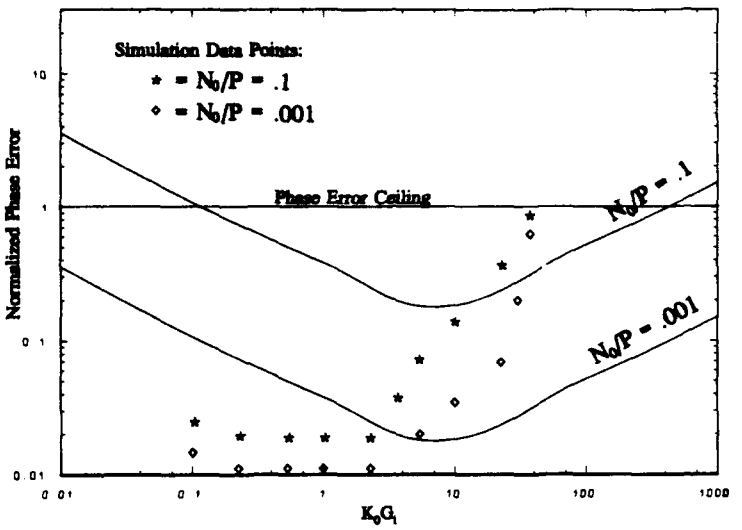


FIGURE 25 MTLL SIMULATION RESULTS WITH $G_2 = 50$

5.5 SUGGESTED OPERATING POINTS

For the MTLL using second-order loops, operating points for maximum loop performance, in the presence of AWGN, can be determined from the plots. The loop appears to operate best at $G_2 = 40$ and $K_0G_1 = 2$ where the maximum phase error is approximately 0.015 for $N_0/P = 0.1$ and 0.01 for $N_0/P = 0.001$. Another good operating point is $G_2 = 50$ and $K_0G_1 = 1$.

5.6 CHAPTER SUMMARY

The equations used to analyze the loop in the presence of AWGN were developed. Also, the method used to determine loop performance was discussed. The theoretical phase error ceiling and maximum phase error curves were derived and plotted on each graph. The results showed that the loop maintained lock at lower vales of phase error than predicted by theory, but only up to a threshold value of $K_0G_1 = 20$. Values of K_0G_1 greater than the threshold region resulted in loop instability where cycle slips occurred. While simulation trends corresponded to that predicted by theory, the threshold effect was not predicted and limits the range of K_0G_1.

VI. COMBINING THE DLL AND THE MTLL

6.1 CHAPTER OVERVIEW

This chapter documents the combined DLL/MTLL performance. The two tracking loops were joined together to determine their overall performance. Section 6.2 develops the equations used to analyze the DLL/MTLL combination when it is synchronized with the input signal for data recovery. In Section 6.3, the model used to emulate the combination of the DLL and MTLL models is introduced. Section 6.4 displays the results of the simulations with the input signal degraded with AWGN for different values of N_0/P, G_2, K_cG_1, and K_0G_1. Suggested operating points are listed in Section 6.5. Finally, Section 6.6 concludes the chapter with a summary.

6.2 DLL/MTLL ANALYSIS FOR DATA RECOVERY

The block diagram of the DLL/MTLL is shown in Figure 26. This analysis will determine the precise point to transfer the signal and recover the data modulated on a GPS input signal in the absence of noise. After the input signal, s(t) (Equation 24), is mixed with the output of the VCO, v(t) (Equation 25), the quadrature arm of the loop becomes:

$$s_Q(t) = \sqrt{PP_v}\, K_m\, m(t)g(t)\left\{\cos[\Theta(t) - \Theta_v(t)] - \cos[2\omega_0 t + \Theta(t) + \Theta_v(t)]\right\} \qquad (43)$$

Since the quadrature signal more closely resembles the PN sequence for the chosen values of s(t) and v(t), this is the signal transferred to the DLL for PN synchronization. The in-phase signal is

just a series of phase shifts which the DLL has trouble synchronizing with the incoming PN sequence. Note when $\Theta(t) = \Theta_v(t)$, the MTLL is synchronized with the carrier signal.

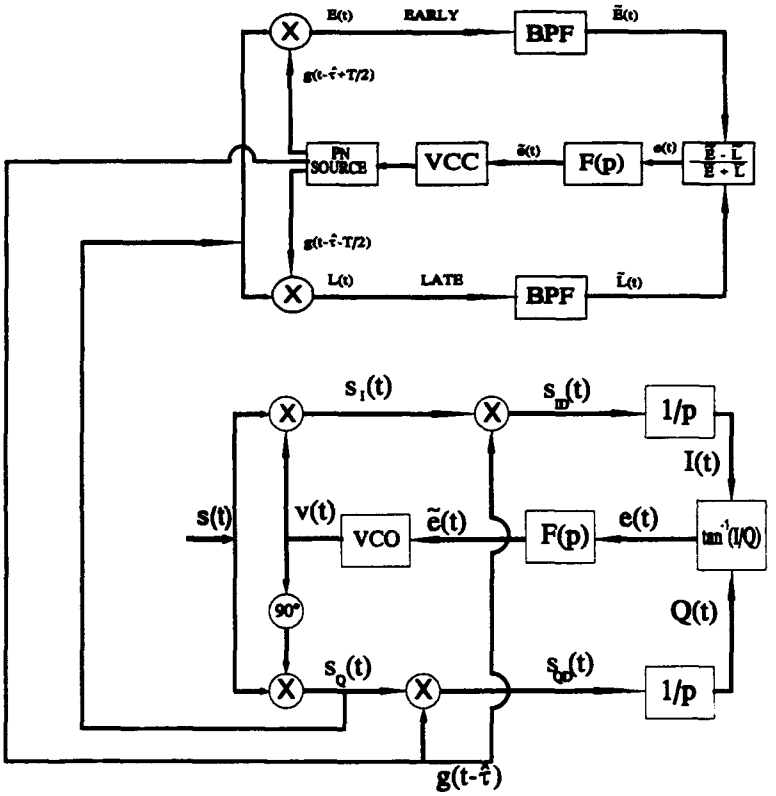


FIGURE 26 DLL/MTLL COMBINATION

When synchronization occurs, the signal to the DLL becomes:

$$s_Q(t) = \sqrt{PP_v}\, K_m\, m(t)g(t)\{1 - \cos[2(\omega_0 t + \Theta(t))]\} \tag{44}$$

where K_m is the gain of the first mixer. Similarly, the in-phase arm of the loop (Equation 26) becomes:

$$s_I(t) = \sqrt{PP_v}\, K_m\, m(t)g(t)\{\sin[2(\omega_0 t + \Theta(t))]\} \tag{45}$$

The on-time signal from the DLL is denoted $g(t - \tau_e)$ where τ_e is the time offset between the incoming PN sequence and the loop's estimate of it ($\tau_e = \tau - \hat{\tau}$). By mixing $g(t - \tau_e)$ with $g(t)$ in the second mixer, it yields the equation:

$$g(t)g(t-\tau_e) = R_g(\tau_e) + n_s(t, \tau_e) \tag{46}$$

where: $R_g(\tau_e) = E[g(t)g(t - \tau_e)]$, the mean of the product $g(t)g(t - \tau_e)$, also known as the autocorrelation function of $g(t)$.

$n_s(t, \tau_e) = g(t)g(t - \tau_e) - R_g(\tau_e)$, the product term with the mean subtracted out, also called the self-noise, which can be neglected since the loop bandwidth is much less than the PN sequence chip rate [SPI63].

When the on-time PN sequence from the DLL is synchronized with the incoming PN sequence, then $\tau_e = 0$. Now Equation 46 becomes $R_g(0) = g(t)g(t) = 1$ for all time.

Now that the PN sequence is removed from the input signal, the signal is despread and Equation 44 becomes:

$$s_{QD}(t) = \sqrt{PP_v}\, K_m K_g m(t)\{1 - \cos[2(\omega_0 t + \Theta(t))]\} \tag{47}$$

where K_g is the gain of the second mixer.

Similarly, after the second mixer in the in-phase arm of the loop, Equation 45 becomes:

$$s_{ID}(t) = \sqrt{PP_v}\, K_m K_g m(t)\{\sin[2(\omega_0 t + \Theta(t))]\} \tag{48}$$

Figure 27 shows all of the signals to this point with no modulated data, m(t), on the input signal and $P = P_v = 1$. Signal 1 (S1) represents the BPSK signal, s(t), into the DLL/MTLL combination. S2 represents the signal in the quadrature arm of the loop after the first mixer, $s_Q(t)$, which is sent to the DLL. The signal in the quadrature arm of the loop after the second mixer, $s_{QD}(t)$, is represented by S3 (note the PN sequence is removed). S4 represents the signal in the in-phase arm of the loop after the first mixer, $s_I(t)$. The signal in the in-phase arm of the loop after the second mixer, $s_{ID}(t)$, is represented by S5 (note the PN sequence is removed).

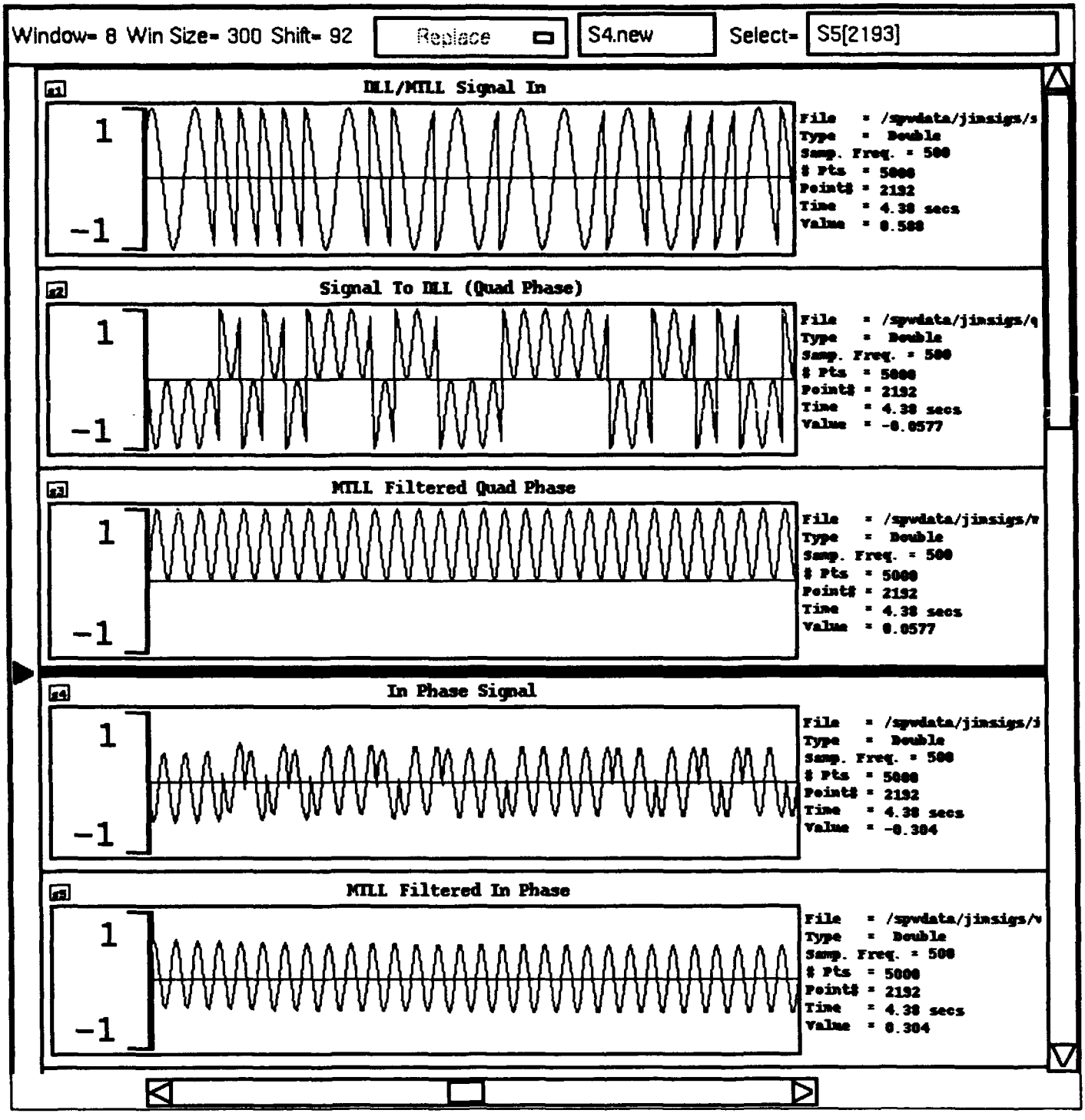


FIGURE 27 DLL/MTLL SIGNALS

After the signal is filtered in the quadrature arm, Equation 47 becomes:

$$Q(t) = \sqrt{PP_v}K_m K_g \tilde{m}(t) \tag{49}$$

since the cosine term is at twice the carrier frequency and is removed by the filter. Similarly, after the filter in the in-phase arm, Equation 48 becomes:

$$I(t) = 0 \tag{50}$$

since the sine term is at twice the carrier frequency and is removed by the filter. Therefore, the error signal, e(t), will be equal to zero since $e(t) = \tan^{-1}[I(t)/Q(t)]$. Now that the carrier signal has also been stripped from the input signal, the data, m(t), can now be recovered from the quadrature arm of the loop, Q(t).

6.3 DLL/MTLL Model

The DLL and the MTLL enable a receiver to lock onto and track the PN code and carrier signal contained within a GPS satellite navigation signal. The two loops must be able to track the GPS signal through noise and frequency-shift degradations. Figure 28 depicts the model used to perform that function. The INPUT SIGNAL block generates a PN sequence which is passed through a -5 BULK DELAY block to emulate an "ontime" PN sequence. The MTLL INPUT SIGNAL block generates a sinusoid emulating a carrier wave. A T_> = T_ON block controls when frequency steps and ramps can be added to the two input signal blocks.

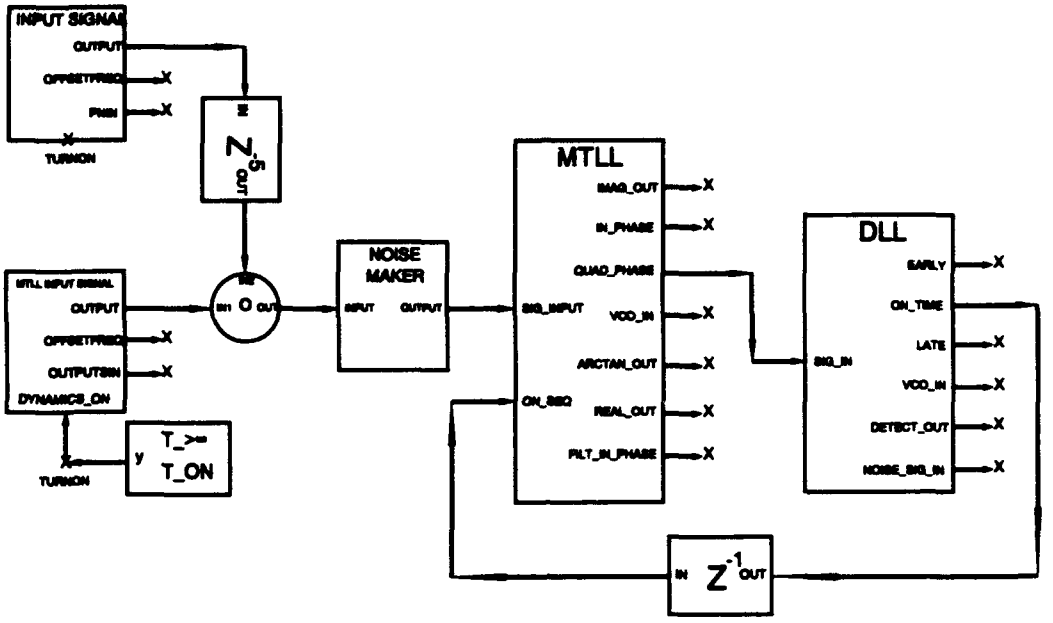


FIGURE 28 COMBINED DLL/MTLL

The two input signals are first passed to a mixer. At the mixer, the two input signals are multiplied together to form the receiver input signal. The input signal is now a BPSK signal with 180° phase shifts indicating the PN sequence. Also, the receiver input signal is passed through a NOISE MAKER block where AWGN is added. Then, the input signal is passed to the MTLL block. The MTLL block will strip the carrier signal off of the input signal by mixing it with a locally-generated carrier signal. The quadrature phase arm of the MTLL passes the signal to the DLL. The DLL will lock onto the incoming PN sequence and pass the on-time local PN sequence back to the MTLL The MTLL block will use that PN sequence to remove the incoming PN sequence from the input signal. This allows the MTLL to synchronize with the incoming carrier signal. A -1 UNIT DELAY block is added due to the software restriction requiring a delay within any feedback loop. Now, each block will be examined in detail.

6.3.1 INPUT SIGNAL BLOCK. The INPUT SIGNAL block is the same one used to generate the incoming PN sequence for the DLL. See Figure 4 for a depiction of the internal components and Section 2.4.1 for a detailed description of the block's operation. For the purposes of this simulation, the noise was not added to the PN sequence in this block and was turned off.

6.3.2 MTLL INPUT SIGNAL BLOCK. The MTLL INPUT SIGNAL block is the same one used to generate the carrier signal (sine wave) for the MTLL. See Figure 20 for a depiction of the internal components and Section 4.4.1 for a detailed description of the block's operation. For the purposes of this simulation, the noise was not added to the carrier signal in this block and was turned off.

6.3.3 NOISE MAKER BLOCK. The NOISE MAKER block is the same one used in the DLL and the MTLL to add noise to an input signal. See Figure 7 for a depiction of the internal components and Section 2.4.1 for an explanation of its operation.

6.3.4 MTLL BLOCK. The MTLL block is used to remove the carrier and PN sequence from the input signal in order to recover the data. Also, it sends an input signal to the DLL, and in return, receives the ontime PN sequence from the DLL in order to remove the incoming PN sequence and despread the input signal. Figure 29 displays the components used to model the MTLL block. The signal sent to the DLL is taken from the QUAD_PHASE point on the quadrature arm of the loop. The ontime PN sequence from the DLL is received at mixers in both arms of the loop. For a detailed description of the MTLL and its components, see Chapter 4.

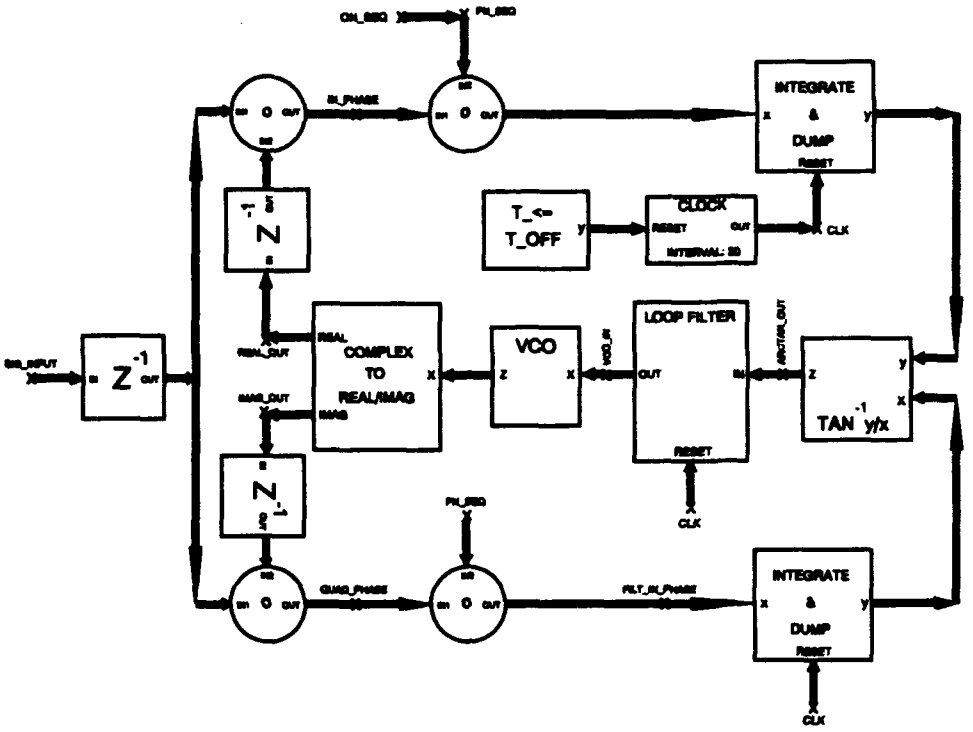


FIGURE 29 MTLL CONFIGURATION

6.3.5 DLL BLOCK. The DLL block is used to lock onto the incoming PN sequence and assist the MTLL in removing that PN sequence from the input signal. Figure 30 displays the components used to model the DLL. Upon receiving the QUAD_PHASE signal from the MTLL at the SIG_IN point, the DLL will lock onto the incoming PN sequence contained within the signal. When the DLL has locked onto the incoming PN sequence, the ontime signal is sent to the MTLL where it is multiplied with the in-phase and quadrature phase arms of the loop to despread the signal and remove the PN sequence from the input signal. For a detailed description of the DLL and its components, see Chapter 2.

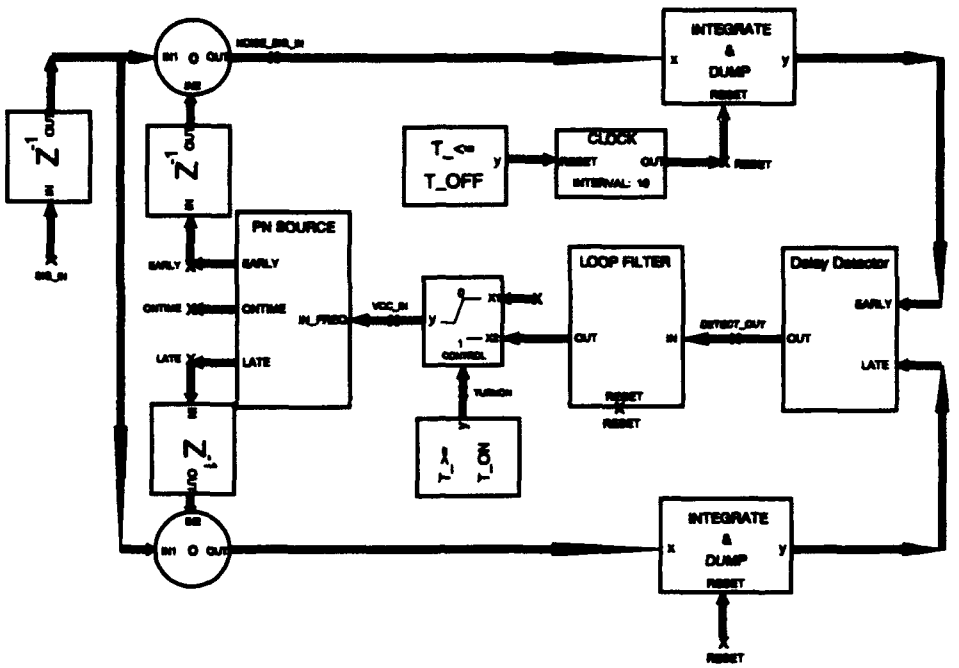


FIGURE 30 DLL CONFIGURATION

6.4 SIMULATION RESULTS

The performance of the DLL/MTLL model, with an input signal degraded only by AWGN, had mixed results. The performance of the DLL was better than predicted by theory, however, the performance of the MTLL was worse than predicted by theory. Both loops encountered threshold effects which limited their range of K_cG_1 and K_0G_1. The threshold effect maybe due to the loops reaching a feedback instability point where errors are compounded causing the loops to drift further from lock. The procedures used to determine loop performance of the DLL/MTLL model are the same ones discussed in Sections 3.3 and 5.3 respectively. Each figure was plotted by using 10,000 simulator iterations with varying noise seeds. Both loops were initially tested as first-order loops then increased to second-order.

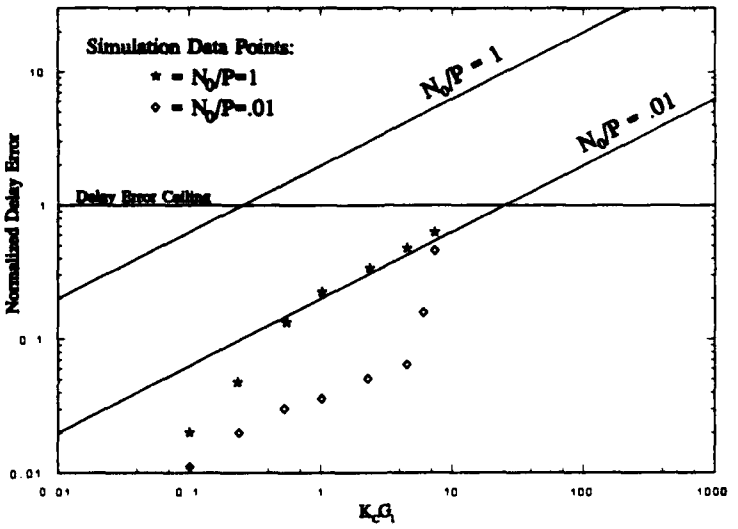


FIGURE 31 DLL SIMULATION RESULTS WITH $G_2 = 0$

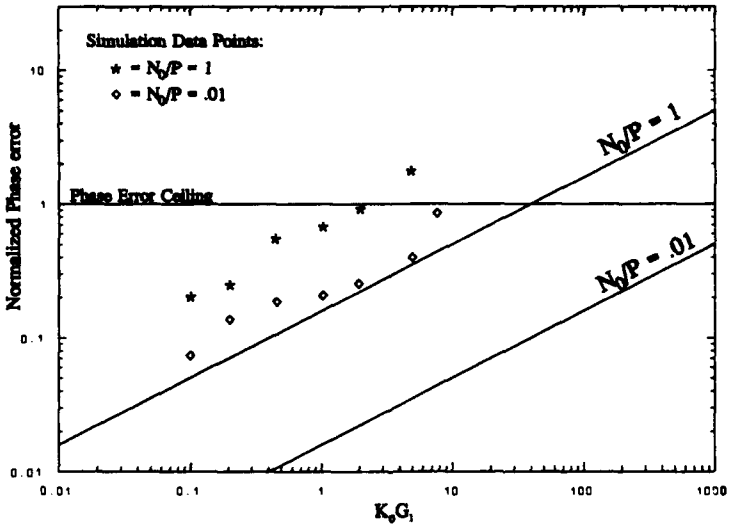


FIGURE 32 MTLL SIMULATION RESULTS WITH $G_2 = 0$

First, N_0/P was set to values of 1 and 0.01. The G_2 parameter of each loop was set to zero to simulate a first-order loop. Figure 31 displays the results of DLL performance. The DLL

error performance remained below levels predicted by theory, and increased delay error with increased loop gain and increased N_0/P is evident. Figure 32 displays the results of MTLL performance. The MTLL performed worse than predicted by theory. While the trend of increased phase error with increased loop gain is evident, the phase errors are greater than the phase error theoretical performance curves.

Next, the results of increasing both loops to second-order and setting $G_2 = 1$ for the DLL and $G_2 = 40$ for the MTLL were plotted. The values of N_0/P were increased to 0.1 and 0.001. Again the performance of the DLL was better than the performance of the MTLL. The trend of increased delay error with increased N_0/P is evident. As values of K_cG_1 are increased the delay error trend follows along theoretical curves. As values of K_0G_1 are increased the phase error trend is above theoretical curves. Both loops experienced a threshold effect above $K_cG_1 = K_0G_1 = 7$. Figure 33 depicts the results of the DLL performance. Figure 34 depicts the results of MTLL performance.

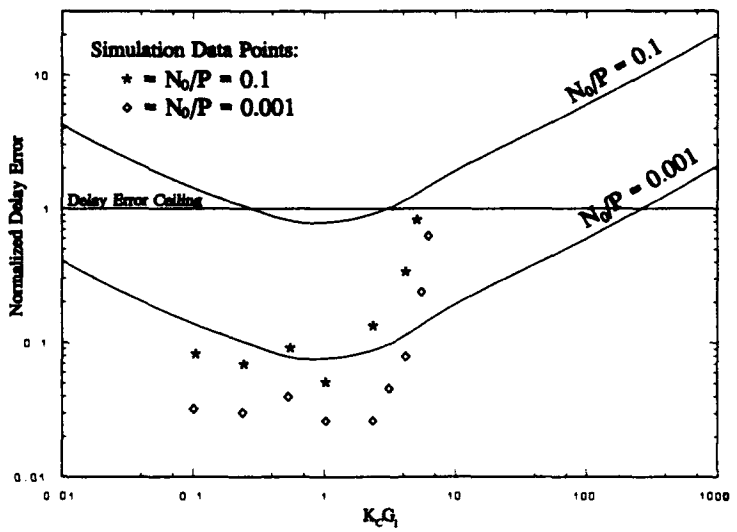


FIGURE 33 DLL SIMULATION RESULTS WITH $G_2 = 1$

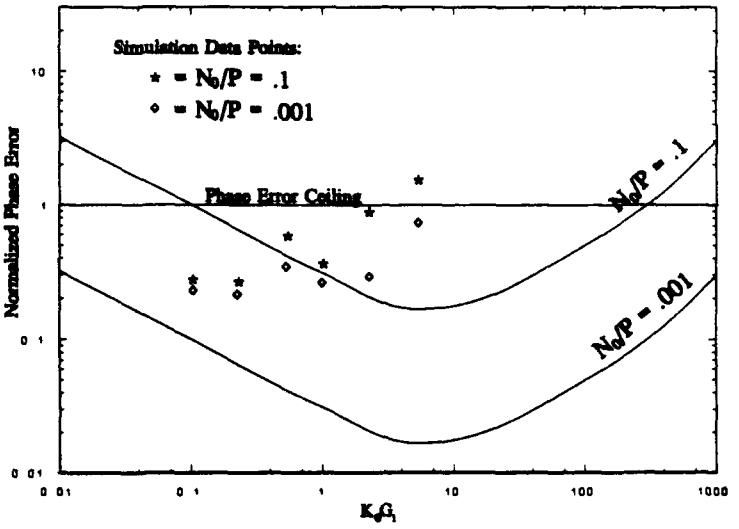


FIGURE 34 MTLL SIMULATION RESULTS WITH G_2 = 40

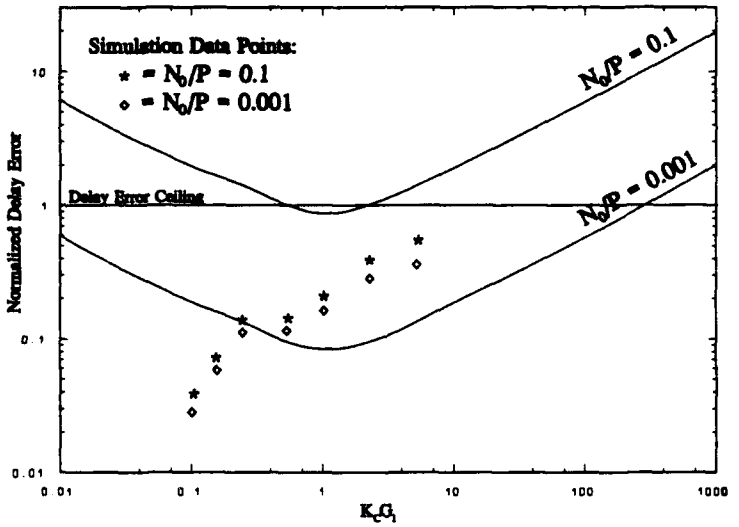


FIGURE 35 DLL SIMULATION RESULTS WITH G_2 = 2

Then, the results of $G_2 = 2$ for the DLL and $G_2 = 50$ for the MTLL were plotted. From all the plots, the threshold effect can be clearly seen occurring around $K_0G_1 = K_cG_1 = 6$. A trend

of increased delay error with increased N_0/P and K_cG_1 is evident, however, the performance of the loop for values less than one does not follow theoretical curves [Fig 35]. A trend of increased phase error with increased N_0/P is apparent, however, the trend in increased phase error with increased K_0G_1 is above theoretical curves and vaguely follows theory [Fig 36]. Cycle slips occurred in the threshold region.

Finally, both loops were set at their optimum operating points, determined from Chapters 3 and 5, with $G_2 = 1$ and $K_cG_1 = 1$ for the DLL and $G_2 = 40$ with $K_0G_1 = 2$ for the MTLL. For $N_0/P = 0.1$, a delay error of 0.1 was produced from the DLL and a phase error of 0.73 was produced from the MTLL. For $N_0/P = 0.001$, a delay error of 0.03 was produced from the DLL and a phase error of 0.33 was produced from the MTLL.

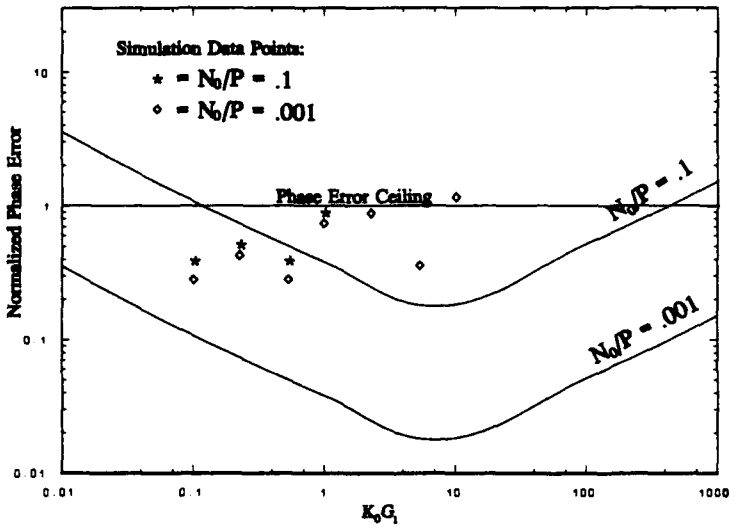


FIGURE 36 MTLL SIMULATION RESULTS WITH $G_2 = 50$

6.5 SUGGESTED OPERATING POINTS

For the DLL/MTLL combination using second-order loops, operating points for maximum loop performance, in the presence of AWGN, can be determined from the plots. For the DLL,

the loop appears to operate with the lowest delay error at $G_2 = 1$ and $K_cG_1 = 1$ where the maximum delay error is approximately 0.05 for $N_0/P = 0.1$ and 0.03 for $N_0/P = 0.001$. The delay error values are comparable to delay error values for the individual loop obtained in Chapter 3. For the MTLL, the loop appears to operate with the lowest phase error at $G_2 = 40$ and $K_0G_1 < 0.05$ where the maximum phase error is approximately 0.3 for $N_0/P = 0.1$ and 0.25 for $N_0/P = 0.001$. The phase error values are not comparable to phase error values for the individual loop obtained in Chapter 5.

6.6 Chapter Summary

This chapter described the equations derived for analysis of data recovery and discussed the software model used to simulate the DLL/MTLL. Simulations demonstrating DLL/MTLL performance in the presence of AWGN were plotted and suggested operating points were addressed. The results indicate a better performance for the DLL than predicted by theory and a worse performance for the MTLL. This was probably due to the threshold effect occurring sooner for loop gain values in the DLL than loop gain values of the MTLL obtained in the individual loop analyses. A threshold effect not predicted by theory occurred at values above $K_cG_1 = K_0G_1 = 7$ for lower loop values of G_2 and at values of $K_cG_1 = K_0G_1 = 6$ for higher loop values of G_2. The threshold effect limits the range of K_cG_1 and K_0G_1 due to cycle slips occurring in this region. The threshold effect may be due to the loops reaching a feedback instability point where errors are compounded causing the loops to drift further from lock.

VII. CONCLUSION AND RECOMMENDATIONS

7.1 SUMMARY

This thesis analyzed two tracking loops for GPS receiver design. A code tracking loop, the Delay-Lock Loop (DLL), a carrier tracking loop, the modified Tanlock Loop (MTLL), and a combined Delay-Lock/modified Tanlock Loop (DLL/MTLL) were investigated. First, the loops were analyzed stand-alone assuming that the other loop was operating perfectly in each case. Then, the two loops were combined and the overall tracking performance of the GPS receiver was investigated. Simulations were performed and the results plotted to determine the performance of the loops in comparison to theoretical predictions. Theoretical performance of the DLL and MTLL tracking an input signal degraded with AWGN was predicted.

For the DLL, equations were presented from prior work which provided an analysis of the loop. The characteristic curve of the loop delay detector was shown to be exactly linear when the delay error is restricted to less than half a chip duration. The DLL model presented is a modification of an earlier model which increases loop order and stability. The modifications included routing the clock signal into the loop filter, which increased the loop filter to first-order, changing all INTEGRATOR blocks to INTEGRATE & DUMP blocks, which increased loop stability, and adjusting the integration cycle to integrate over one chip. Other modifications included an input UNIT DELAY block to compensate for the UNIT DELAY blocks contained within the loop arms, and a timer to control the first integrate and dump cycle by initially resetting the clock. Adding noise to the input signal led to the presentation of equations plotting maximum delay theoretical curves for different values of N_0/P, G_2, and K_cG_1. General trends in loop performance were evident when simulations were performed and plotted. As input noise

increased, the delay error increased, and as loop gain increased, the simulations followed along theoretical curves. The performance of the loop was at or below theory until reaching a threshold region where cycle slips occurred.

For the MTLL, equations were presented from prior work which provided an analysis of the loop. The arctangent phase detector was shown to be exactly linear with a period of 2π. The MTLL model presented is a modification of an earlier model which increases loop order and stability. The modifications include routing the clock signal into the loop filter, which increases the loop filter to first-order, changing all INTEGRATOR blocks to INTEGRATE & DUMP blocks, which increases loop stability, and adjusting the integration cycle to integrate over one period. Other modifications include an input UNIT DELAY block which reduces an initial phase offset, and a timer to control the first integrate and dump cycle by initially resetting the clock. Adding noise to the input signal led to the presentation of equations plotting maximum delay theoretical curves for different values of N_0/P, G_2, and K_0G_1. General trends in loop performance were evident when simulations were performed and plotted. As input noise increased, the delay error increased, and as loop gain increased, the simulations followed along theoretical curves. The performance of the loop was at or below theory until reaching a threshold region where cycle slips occurred.

For the DLL/MTLL, equations were developed which demonstrated the ability of the loop to recover a data signal prior to any noise degrading the loop. The DLL/MTLL model presented is a combination of the modified DLL and MTLL models. Noise was added to the input signal after the PN sequence was modulated onto the carrier signal. The input to the DLL was taken from the quadrature arm of the loop after the input signal was mixed with the VCO output. It was determined through mathematical analysis that the data can be recovered after the quadrature phase

signal is filtered. Simulations were performed using theoretical curves for the individual loops. General trends in loop performance were evident for different value of N_0/P, G_2, K_cG_1, and K_0G_1.

Although loop performance of the DLL and MTLL performed below theory individually, they did not perform as expected when combined. The performance of the DLL in the DLL/MTLL model was better than predicted by theory, however, the performance of the MTLL was worse than predicted by theory. Since the threshold region of the DLL occurs prior to the threshold region of the MTLL, this fact may have resulted in the poor performance of the MTLL when combined with the DLL. Each loop, individual or combined, encountered a threshold effect not predicted by theory which limited their range. The threshold effect maybe due to the loops reaching a feedback instability point where errors are compounded causing the loops to drift further from lock.

7.2 Conclusions/Lessons Learned

7.2.1 The Delay Lock Loop. By adjusting the integration cycle to integrate over one chip, an improvement in loop performance was made over the previous model. By changing to INTEGRATE & DUMP blocks, loop stability was improved. Whenever the locally generated PN sequence was too far advanced of the incoming PN sequence, a small frequency was input to the VCC to slow down the local PN sequence. If the frequency requested was too small, SPW output a warning message and set the frequency input to 1e-200 Hz (which is approximately zero). The frequency output from the function generator inside the VCC is equal to the initial frequency plus the input frequency. Thus, the local PN sequence frequency would not decrease and continue to stay advanced of the incoming PN sequence. This error occurred around the region of the threshold effect. After the INTEGRATOR blocks were changed to INTEGRATE & DUMP

blocks, this warning did not reappear; however, the error was the same. Only the RAM method of PN sequence generation was investigated since it is the nominal method performed in a GPS receiver. An apparent threshold effect limited the range of $K_c G_1$ during simulations. No theoretical derivations can account for this effect. The software restriction requiring delays within a feedback loop probably had an effect on the delay error. However, the magnitude of the effect was not investigated.

7.2.2 THE MODIFIED TANLOCK LOOP. By reducing an initial phase offset error, and adjusting the integration cycle to integrate over one period, the performance of the loop was increased over the previous model. By changing to INTEGRATE & DUMP blocks as filters, loop stability improved and the locally generated carrier signal more closely resembled a sinusoid for second-order loops. An apparent threshold effect limited the range of $K_0 G_1$ during simulations. In the threshold region the loop became unstable and did not resemble a sinusoid. This effect was not discovered during theoretical analysis. The software restriction requiring delays within a feedback loop probably had an effect on the phase error. However, the magnitude of the effect was not investigated.

7.2.3 THE COMBINED DLL/MTLL. Analysis of the DLL/MTLL block diagram, in the absence of noise, determined the precise point where the signal to the DLL should be obtained from the MTLL. Also, the exact point where data recovery should take place was discovered (after the filter in the quadrature phase arm of the loop). The performance of the DLL/MTLL should show an improvement for higher SNRs. Using different sampling frequencies for the DLL and MTLL could have obscured loop performance. An apparent threshold effect limited the range

of K_0G_1 and K_cG_1 during simulations. Since the MTLL has a threshold region that occurs at a higher value of K_0G_1 than the K_cG_1 value of the DLL, this could have contributed to the poor performance of the MTLL. The threshold effect may be due to the loops reaching a feedback instability point where errors are compounded causing the loops to drift further from lock. Using linear analysis to define a non-linear system is probably the reason the threshold effect was not predicted by theory.

7.3 Recommendations for Further Research

For the DLL, a new delay detector using the normalized magnitude of the delay error should be investigated [OUL84:25]. A brief analysis determined that the new delay detector, where $e(t) = (|E| - |L|)/(|E| + |L|)$, showed an improvement over the current delay detector. Increase the PN sequence to the nominal GPS PN sequence length for a C/A code of 1023 chips by adding another PN sequence generator. Increase the operating frequency of the PN sequence to the nominal GPS frequency of a C/A code. Any increase in operating frequency will change the delay blocks required to maintain the half chip offset. By changing the delays to variable delays, the loop can maintain the half chip offset dynamically. For the MTLL, the operating frequency of the carrier signal should be increased to the nominal GPS carrier frequency. Lower values of SNRs should also be investigated. To implement lower values of SNR, higher values of G_2 will be required.

For the DLL/MTLL, the poor performance of the combined loops needs further study. The sampling frequencies of both loops should be increased with the knowledge that the amount of oversampling is limited due to the high operating frequencies required to simulate a GPS receiver. Data should be modulated on the input signal to investigate data recovery and the effects

of a Doppler shift needs to be investigated. Increasing the number of simulations for each data point will smooth out data plots. Increasing the number of sampling iterations will increase the time window and give a more accurate picture of loop performance. Merging the DLL/MTLL with the acquisition loop would complete the GPS receiver tracking design.

[BAL64] Balodis, M. "Laboratory Comparison of TANLOCK and Phaselock Receivers, *Conference Record, National Telemetry Conference, Paper 5-4*: Piscataway, NJ: IRE, 1964.

[CHO87] Cho, W. D. and C. K. Un. "On Improving the Performance of a Digital Tanlock Loop," *Proceedings of the IEEE, 75*: 520-522 (April 1987).

[COO86] Cooper, George R. and Clare D. McGillem. *Modern Communications and Spread Spectrum.* New York: McGraw-Hill Book Company, 1986.

[GAU91] Gaudenzi, Riccardo De and Marco Luise. "Decision-Directed Coherent Delay-Lock Tracking Loop for DS-Spread Spectrum Signals," *IEEE Transactions on Communications, 39:* 758-765 (May 1991).

[GIL66] Gill, Walter J. "A Comparison of Binary Delay-Lock Tracking-Loop Implementations," *IEEE Transactions on Aerospace and Electronic Systems, AES-2* 415-424 (July 1966).

[HIR92] Hird, James A. "Analysis and Simulation of Modified Tanlock and Delay Lock Loops for GPS Receiver Design." MS Thesis, AFIT/GE/ENG/92D-20. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1992.

[LEE82] Lee, Jae Chon and Chong Kwan Un. "Performance Analysis of Digital Tanlock Loop," *IEEE Transactions on Communications, 30*: 2398-2411 (October 1982).

[LIN72] Lindsey, William C. *Synchronization Systems in Communication and Control.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1972.

[OUL84] Ould, Peter C. and Robert J. VanWechel. "All-Digital GPS Receiver Mechanization," *Papers published in Navigation.* 25-35. Washington D.C.: The Institute of Navigation, 1984.

[ROB62] Robinson, L. M. "TANLOCK: A Phase-Lock Loop of Extended Tracking Capability," *Proceedings of IRE Convention on Military Electronics*: 396-427. Piscataway, NJ: IRE, 1962.

[SCH92] Schmitz, Ronald E. "Acquisition of Direct Sequence Spread Spectrum Signals using Digital Signal Processing Techniques." MS Thesis, AFIT/GE/ENG/92D-35. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1992.

[SIM77] Simon, Marvin K., "Noncoherent Pseudonoise Code Tracking Performance of Spread Spectrum Receivers," *IEEE Transactions on Communications, 3:* 327-345 (March 1977).

[SPI63] Spilker, J. J., Jr. "Delay-Lock Tracking of Binary Signals," *IEEE Transactions on Space Electronics and Telemetry, Set-9* : 1-8 (March 1963).

[SPI80] Spilker, J. J., Jr. "GPS Signal Structure and Performance Characteristics," *Papers Published in Navigation, Vol 1.* Washington D.C.: Institute of Navigation, 1980.

[YOS80] Yost, Richard A. and Robert W. Boyd. "A Modified PN Code Tracking Loop: Its Performance and Implementation Sensitivities," *Proceedings National Tele-Communications Conference*: 61.5-1 - 61.5-5. Houston, Tx: IEEE, December 1980.

[YOS82] Yost, Richard A. and Robert W. Boyd. "A Modified PN Code Tracking Loop: Its Performance Analysis and Comparative Analysis," *IEEE Transactions on Communications COM-30*: 1027-1036 (May 1982).

Vita

Captain George D. Harris was born in Orlando, Florida on February 20, 1958. He graduated from Hillcrest High School in 1976 and entered the U.S. Navy. He spent six years working on a guided missile destroyer where he was supervisor of an engine-room. After his discharge in 1982, he attended the University of South Carolina where he received his Bachelor of Science Degree in Electrical Engineering in 1986. He attended Officer Training School in 1987 where he received his commission as a Second Lieutenant in the U.S. Air Force. He attended Undergraduate Space Training School before arriving at the Second Satellite Operations Squadron located at Falcon AFB, Colorado. He worked as a GPS Crew Commander until transferring to the Standardization and Evaluation branch in 1990 where he worked as a Satellite Operations/Crew Commander Evaluator. In 1992, he was assigned to the Air Force Institute of Technology to earn his Master of Science Degree in Electrical Engineering.

Permanent Address: 411 Woodknoll Dr.

W. Carrollton, OH 45449

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1993	Master's Thesis

4. TITLE AND SUBTITLE
Analysis and Simulation of a GPS Receiver Design Using Combined Delay-Lock and Modified Tanlock Loops

5. FUNDING NUMBERS

6. AUTHOR(S)
George D. Harris, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/93D-13

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Todd Jenkins
WL/AAAI-3
Wright Laboratory
WPAFB OH, 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The purpose of this thesis was to investigate the performance of two types of tracking loops used in Global Positioning System (GPS) receivers. The first loop, the Delay-Lock Loop (DLL), is responsible for maintaining synchronization with the received PN sequence. The second loop, the Modified Tanlock Loop (MTLL), is responsible for maintaining synchronization with the carrier signal. The performance of the two loops is investigated first separately then their performance is evaluated when operated together. This thesis is an investigation on the ability of these two loops to overcome corruption of the input signal due to noise. Expanding the dynamic operating range of these loops can significantly improve GPS receiver operation. Results indicate the performance of the loops was better than theoretical predictions by maintaining lock across a wide range of loop gains and SNRs. However, when the loops were combined, the loops did not perform as predicted by theory. All simulations display phenomena which was not present in the theoretical predictions.

14. SUBJECT TERMS	15. NUMBER OF PAGES
GPS, Spread Spectrum, Delay Lock Loop, Tanlock Loop, Tracking, Synchronization	93
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL